MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

du-CBA: VERİDEN HABERSİZ VE ARTIRIMLI SINIFLANDIRMAYA DAYALI BİRLİKTELİK KURALLARI ÇIKARMA MİMARİSİ

BÜŞRA BÜYÜKTANIR

YÜKSEK LİSANS TEZİ
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Yüksek Lisans Programı

DANIŞMAN
Doç. Dr. Kazım YILDIZ

EŞ-DANIŞMAN
Dr. Öğr. Üyesi Eyüp Emre ÜLKÜ

İSTANBUL, 2022

MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

du-CBA: VERİDEN HABERSİZ VE ARTIRIMLI SINIFLANDIRMAYA DAYALI BİRLİKTELİK KURALLARI ÇIKARMA MİMARİSİ

BÜŞRA BÜYÜKTANIR
523618018

YÜKSEK LİSANS TEZİ
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Yüksek Lisans Programı

DANIŞMAN
Doç. Dr. Kazım YILDIZ

EŞ-DANIŞMAN
Dr. Öğr. Üyesi Eyüp Emre ÜLKÜ

İSTANBUL, 2022

ÖNSÖZ

Tez çalışmamın araştırılmasında ve yürütülmesinde her daim destek olan, her fırsatta yardımını esirgemeyen değerli danışmanım Doç.Dr. Kazım YILDIZ'a, eşdanışmanım Dr. Öğr. Üyesi Eyüp Emre ÜLKÜ'ye,

Hayatımın her alanında olduğu gibi yüksek lisans ve tez çalışmam boyunca desteğini, sabrını ve sevgisini esirgemeyen kıymetli eşim Tolga BÜYÜKTANIR'a, bugünlere gelmemde büyük pay sahibi olan değerli annem Fatma TEKE'ye ve manevi desteklerini hiçbir zaman eksik etmeyen canım kardeşlerim Hatice Nur ÖZER'e, Halil İbrahim ÖZER'e ve Oğuz Han ÖZER'e

Sonsuz teşekkürlerimi sunuyorum.

Son olarak bu çalışmamı güzel kızlarım Zeynep Şeyda'ya ve Eslem Esra'ya armağan ediyorum.

TEMMUZ, 2022 **Büşra BÜYÜKTANIR**

İÇİNDEKİLER

SAYFA

ÖNSÖZ i
İÇİNDEKİLER ii
ÖZET iv
ABSTRACT vi
SEMBOLLER viii
KISALTMALAR ix
ŞEKİL LİSTESİ xi
TABLO LİSTESİ xii
1. GİRİŞ 1
1.1. Genel Bakış 2
1.1.1. Problemin Tanımı 3
1.1.2. Amaç ve Hedef 4
1.1.3. Ana Katkılar 4
1.1.4. Tez Organizasyonu 4
1.2. Federe Öğrenme 5
1.2.1. Federe Öğrenme Mimarisinin İşleyişi 6
1.2.2. Federe Öğrenme Mimarisinde Veri Gizliliği ve Veri Güvenliği 7
1.2.3. Federe Öğrenme Modelleri 9
1.2.4. Federe Öğrenme Kullanım Alanları 9
1.3. Artırımlı Öğrenme 11
1.4. İlişkisel Sınıflandırma 12
1.4.1. Sınıflandırma 14
1.4.2. Birliktelik Kuralları 14
1.5. Literatür Taraması 17
2. YÖNTEM 20
2.1. Veri Seti 20
2.2. du-CBA 21
2.2.1. du-CBA ile Federe Öğrenme Mimarisi 21
2.2.2. du-CBA Algoritması Çalışma Mantığı 22
3. BULGULAR ve TARTIŞMA 28
3.1. Prototip Uygulama 28
3.2. Deney Ortamı 29
3.3. Performans Ölçütleri 30
3.4. Deney Sonuçları 31
4. SONUÇLAR 41

KAYNAKLAR 43
ÖZGEÇMİŞ 47

ÖZET

du-CBA: VERİDEN HABERSİZ VE ARTIRIMLI SINIFLANDIRMAYA DAYALI BİRLİKTELİK KURALLARI ÇIKARMA MİMARİSİ

Günümüzde nesnelerin interneti, mobil uygulamalar gibi hayatımızı kolaylaştıran, istemcilerin ve sunucuların birlikte çalışması gereken sistemlerin kullanımı artmıştır. Bu alanlarda makine öğrenmesi modeli kullanılması ihtiyaç haline gelmiştir. Ancak istemcilerden verilerin toplanması, sunucuya aktarılması, makine öğrenmesi modeli eğitilmesi ve ardından bu modelin istemcilerde çalışan cihazlara entegre edilmesi bir çok problemi beraberinde getirmektedir. Verilerin istemcilerden sunucuya transferi ağ trafiğine sebep olmakta ve fazla enerji gerektirmektedir. Veriler içerisinde kişisel veriler yer almaktadır. Transfer sırasında kişisel verilerin aktarılması veri mahremiyetini istismar edilebilmektedir. Veri mahremiyetini sağlamak için veri paylaşımının yapılmaması ise veri çeşitliliğini ve büyüklüğünü azalttığından dolayı sunucuda geliştirilecek modelin doğru sonuç vermemesine sebep olmakta, verimliliğini düşürmekte ve bu alandaki makine öğrenmesi çalışmalarını kısıtlamaktadır.

Tez çalışması kapsamında bahsedilen bu problemlere çözüm üretmek için federe öğrenme mimarisi kullanılmaktadır. Mimariye göre her bir istemci, kendi verilerini kullanılarak yerel bir makine öğrenmesi modeli eğitmektedir. Eğitilen modeller sunucuya gönderilmekte ve sunucuda bu modeller birleştirilerek yeni bir model oluşturulmaktadır. Oluşturulan nihai model tekrar istemcilere dağıtılmaktadır. Bu sayede ağ trafiği azaltılmakta, enerji ihtiyacı düşürülmektedir. Veri mahremiyeti ise, bütün veriler yerine tek başına anlamsız olan veriler gönderildiği için korunmaktadır.

Federe öğrenme mimarisi güncel ve geliştirilmeye açık bir alandır. Bu alanda geliştirilecek algoritmalara ihtiyaç duyulmaktadır. Bu çalışmada Veriden Habersiz İlişkili Kurallara Dayalı Sınıflandırma (Data Unaware Classification Based on Association, du-CBA) olarak adlandırılan algoritma geliştirilmiştir. Algoritma federe öğrenme mimarisi için geliştirilmiş ilişkisel sınıflandırma algoritmasıdır. Federe öğrenme ile klasik öğrenme mimarilerini karşılaştırıp başarılarını ölçmek için çalışma kapsamında bir simülasyon ortamı oluşturulmuştur. Simülasyonda du-CBA ve CBA algoritmaları kullanılarak modeller eğitilmiş ve sonuçlar kıyaslanmıştır. Modellerin eğitiminde University of California Irvine (UCI) veri havuzundan alınan beş veri seti kullanılmıştır.

Deneysel sonuçlar ayrı ayrı her bir veri seti için federe öğrenme ile eğitilen modelin, klasik öğrenme ile eğitilen modelle neredeyse aynı doğruluğu elde ettiğini göstermiştir. Model federe öğrenme yöntemi ile eğitildiğinde, eğitim süresinin yaklaşık olarak %70 oranında azaldığı ortaya çıkmıştır. Böylece federe öğrenme mimarisini kullanacak cihazlarda enerji ihtiyacının düştüğü sonucuna varılmaktadır. Ayrıca veri yerine eğitilmiş model merkeze gönderildiği için hem veri mahremiyeti sağlanmış hem de ağ trafiği kayda değer şekilde azalmıştır. Deneysel sonuçlar geliştirilen algoritmanın başarılı sonuçlar ürettiğini göstermektedir.

TEMMUZ, 2022 **Büşra BÜYÜKTANIR**

ABSTRACT

du-CBA: DATA AGNOSTIC AND INCREMENTAL CLASSIFICATION BASED ASSOCIATION RULES EXTRACTION ARCHITECTURE

Today, the use of systems that make our life easier, such as the Internet of Things and mobile applications, and which require clients and servers to work together, has increased. The use of machine learning models in these areas has become a necessity. However, collecting data from the clients, transferring them to the server, training the machine learning model and then integrating this model into the devices running on the clients bring along many problems. The transfer of data from the clients to the server causes network traffic and requires a lot of energy. The data includes personal data. The transfer of personal data during the transfer can exploit data privacy. The lack of data sharing to ensure data privacy, on the other hand, reduces the variety and size of data, causing the model to be developed on the server to not give correct results, reducing its efficiency and restricting machine learning studies in this area.

Federated learning architecture is used to produce solutions to these problems mentioned within the scope of the thesis. According to the architecture, each client trains a local machine learning model using its own data. The trained models are sent to the server and a new model is created by merging these models on the server. The final model created is distributed to the clients again. In this way, network traffic is reduced and energy demand is reduced. Data privacy, on the other hand, is protected as only meaningless data is sent instead of all data.

Federated learning architecture is an up-to-date and open field. Algorithms to be developed in this area are needed. In this study, an algorithm called Data Unaware Classification Based on Association (du-CBA) has been developed. The algorithm is an association classification algorithm developed for federated learning architecture. In order to compare federated learning and classical learning architectures and measure their success, a simulation environment was created within the scope of the study. Models were trained using du-CBA and CBA algorithms in the simulation and the results were compared. Five data sets from the University of California Irvine (UCI) repository were used to train the models.

Experimental results showed that for each data set separately, the model trained with federated learning achieved almost the same accuracy as the model trained with classical learning. When the model is trained with the federated learning method, it has been revealed that the training

time is reduced by approximately 70%. Thus, it is concluded that the energy requirement of the devices that will use the federated learning architecture has decreased. In addition, since the trained model is sent to the center instead of data, both data privacy is ensured and network traffic is significantly reduced. Experimental results show that the developed algorithm produces successful results.

JULY, 2022 **Büşra BÜYÜKTANIR**

SEMBOLLER

N **:** Modellerin eğitiminde kullanılan veri içerisindeki örnek sayı bilgisi

n **:** İstemci sunucu sistemlerinde uçta çalışan bilinmeyen cihaz sayısı

X **:** Kuralın sol tarafını (lhs: Left hand side)

Y **:** Kuralın sağ tarafını (rhs: Right hand side)

KISALTMALAR

ACIM **:** Associative Classification Based on İncremental Mining

CARs **:** Class Association Rules

CBA **:** Classification Based on Association

CBA-CB **:** Classification Based on Association Classifier Building

CBA-RG **:** Classification Based on Association Rule Generation

CMAR **:** Classification Based on Multiple Class-Association Rules

du-CBA **:** Veriden Habersiz İlişkili Kurallara Dayalı Sınıflandırma (Data Unaware Classification Based on Association, du-CBA)

DN **:** Doğru Negatif

DP **:** Doğru Pozitif

E-ACIM **:** Enhanced Associative Classification Based on İncremental Mining Algorithm

ECBA **:** Enhanced Classification Based on Association Rules

FCBA **:** Fast Classification Based on Association Rule

FN **:** False Negative

FP **:** False Positive

GDPR **:** Genel Veri Koruma Yönetmeliği

GSM **:** Global System for Mobile Communications

IoT **:** Internet of Things / Nesnelerin İnterneti

KVKK **:** Kişisel Verileri Koruma Kanunu

lhs : Left Hand Side

MCAR **:** Multiclass Classification Based on Association Rule

M Ö **:** Milattan Önce

M S **:** Milattan Sonra

rhs **:** Right Hand Side

SMC **:** Güvenli Çok Taraflı Hesaplama / Secure Multiparty Computation

TN **:** True Negative

TP **:** True Positive

UCI **:** University of California

WCBA **:** Weighted Classification Based on Association Rules

YN **:** Yanlış Negatif

YP **:** Yanlış Pozitif

ŞEKİL LİSTESİ

SAYFA

Şekil 1.1. Sunucu istemci modeli. 1
Şekil 1.2. Federe Öğrenme yapısı. 3
Şekil 1.3. Federe öğrenme mimarisinin işleyişi. 6
Şekil 1.4. Federe öğrenme mimarisinde veri gizliliği ve güvenliği. 7
Şekil 1.5. Artırımlı öğrenme işleyişi. 12
Şekil 1.6. Makine öğrenmesi yöntemleri için kullanılan öğrenme tipleri. 13
Şekil 1.7. İlişkisel sınıflandırma yöntemi çalışma adımları. 16
Şekil 2.1. Federe öğrenme mimarisi ile model eğitiminde du-CBA kullanımı. 22
Şekil 2.2. (a) du-CBA algoritmasının istemcide çalışma adımları; (b) du-CBA algoritmasının sunucuda çalışma adımları. 23
Şekil 3.1. Benzetim ortamında kıyaslanan klasik makine öğrenmesi yöntemi ve federe öğrenme yönteminin şekil ile gösterimi. 28
Şekil 3.2. CBA ve du-CBA algoritmalarının Car Evaluation veri setini kullanarak 2, 4, 8, 16, 32, 64 ve 128 olarak belirlenen yedi farklı uç sayıları için tek tek modellerin eğitim sürelerinin karşılaştırılması. 32
Şekil 3.3. CBA ve du-CBA algoritmalarının Bank Marketing veri setini kullanarak 2, 4, 8, 16, 32, 64 ve 128 olarak belirlenen yedi farklı uç sayıları için tek tek modellerin eğitim sürelerinin karşılaştırılması. 33
Şekil 3.4. CBA ve du-CBA algoritmalarının Mushroom veri setini kullanarak 2, 4, 8, 16, 32, 64 ve 128 olarak belirlenen yedi farklı uç sayıları için tek tek modellerin eğitim sürelerinin karşılaştırılması. 34
Şekil 3.5. CBA ve du-CBA algoritmalarının Nursery veri setini kullanarak 2, 4, 8, 16, 32, 64 ve 128 olarak belirlenen yedi farklı uç sayıları için tek tek modellerin eğitim sürelerinin karşılaştırılması. 35
Şekil 3.6. CBA ve du-CBA algoritmalarının Adult veri setini kullanarak 2, 4, 8, 16, 32, 64 ve 128 olarak belirlenen yedi farklı uç sayıları için tek tek modellerin eğitim sürelerinin karşılaştırılması. 36

TABLO LİSTESİ

SAYFA

Tablo 1.1. Federe öğrenme mimarisinin uygulandığı alanlar……10

Tablo 2.1. Çalışma kapsamında kullanılan UCI veri setlerinin özellikleri. 20

Tablo 3.1. Karmaşıklık Matrisi…….30

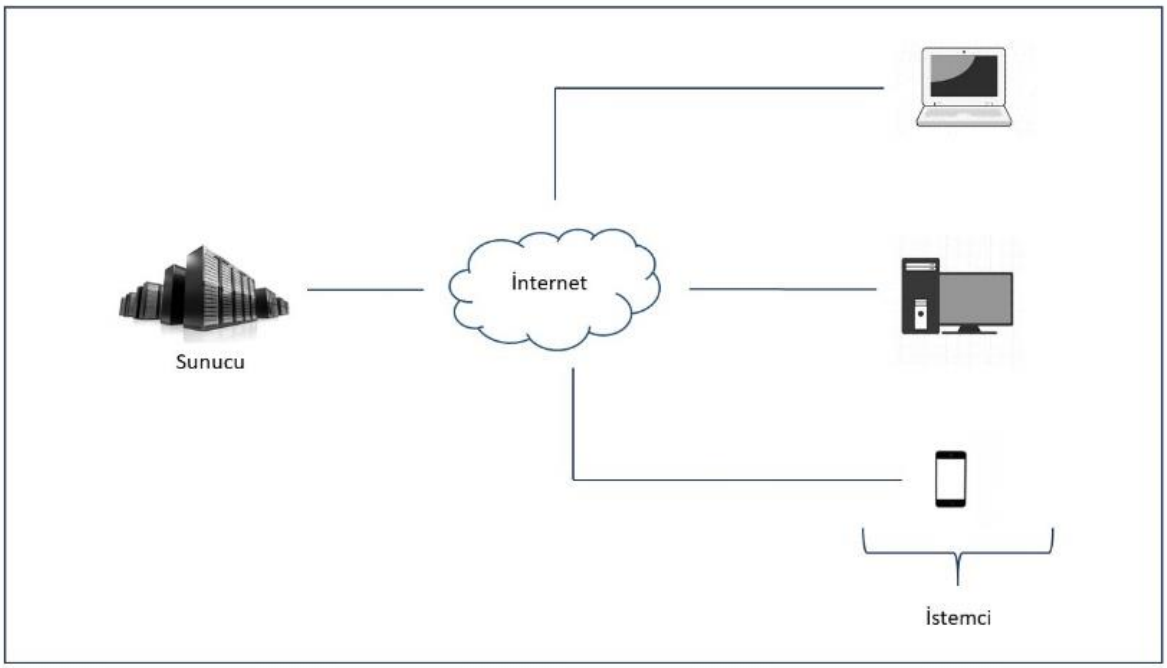
Tablo 3.2. Beş farklı veri kümesi için CBA ve du-CBA algoritmaları ile eğitilen modellerin performans ölçülerinin ortalama değerleri…….38

Tablo 3.3. CBA ve du-CBA algoritmalarının beş farklı veri seti ile eğitildiği modellerin performans ölçütlerine ait standart sapma değerleri……..39

1. GİRİŞ

Dünya üzerindeki tüm canlılar kendilerini ifade edebilmek ve hayatlarını devam ettirebilmek için bilgi alışverişinde bulunmaktadırlar. Yapılan bu alışverişe haberleşme denilmektedir. İnsanlık tarihinin var oluşundan günümüze kadar haberleşme için çeşitli yöntemler denenmiştir. Milattan önce (M Ö) insanlar birbirlerine seslenerek, davul ve duman kullanarak haberleşmeyi sağlamışlardır. Yazının bulunması ile haberleşme bir adım ileri taşınmış ve postacılık sistemi ile haberleşmeye devam edilmiştir. Milattan sonra (M S) elektriğin bulunması ile birçok alanda olduğu gibi haberleşme alanında da gelişmeler meydana gelmiştir. Telgraf, telefon, Global System for Mobile Communications (GSM) teknolojisi, bilgisayar ve internet ile insanlık tarihi boyunca var olan haberleşme, gelişmeye devam etmektedir.

Haberleşmenin gelişmesi sürecinde internet kavramı da hayatımıza girmiştir. İnsanlar ve cihazlar arasında iletişimi sağlayan haberleşme ağına internet denilmektedir. Teknolojinin de gelişmesi ile birlikte internet insan hayatını kolaylaştırmıştır. Bilişimden finansa, eğitimden sağlığa birçok alanda kendini göstermiştir. Günlük iş ve işlemlerin gerçekleştirildiği mobil uygulamalar yaygınlaşmıştır. Nesnelerin de bu iletişim ağına dahil olması ile nesnelerin interneti (IoT) kavramı ortaya çıkmıştır [1]. Bu sayede birden fazla nesne, veri gönderim ve alımı ile haberleşmeyi sağlamıştır.



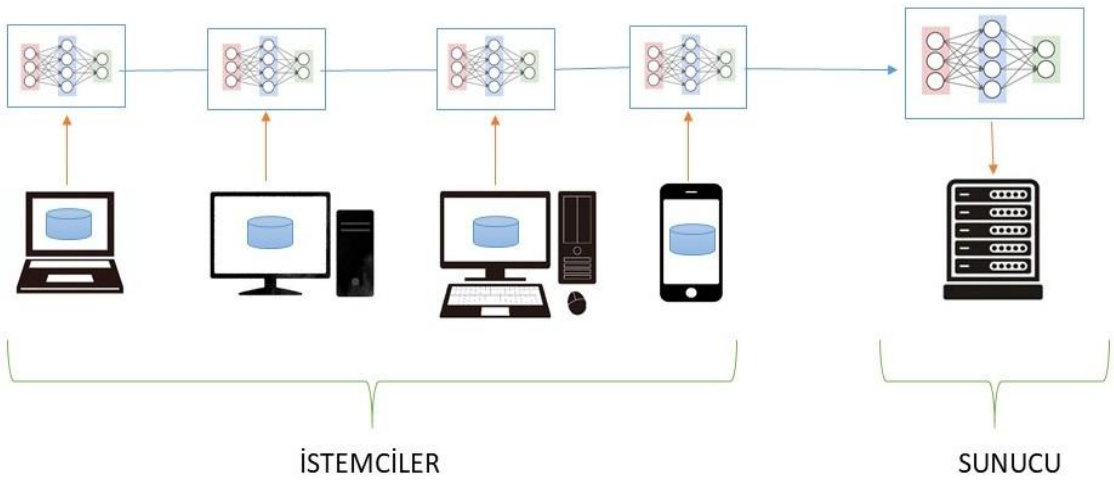
Şekil 1.1. Sunucu istemci modeli.

İnternet, sunucu ve istemci modelinden oluşan bir sistemdir. Haberleşme bu sistem arasında gerçekleşmektedir. Şekil 1.1.'de gösterilen sunucu ve istemci modelinde, çeşitli hizmetleri talep eden programı çalıştıran cihazlara istemci makinesi, bir veya birden fazla gelen talebe cevap veren programı çalıştıran cihaza ise sunucu makinesi denilmektedir.

İnternet teknolojilerinin hayatın her alanına girmesi ile birlikte insanlar günlük yaşantılarında veri üretir hale gelmişlerdir [2, 3]. Bu veriler arasındaki ilişkiyi bulmak ve ortaya çıkartmak için makine öğrenmesi yöntemleri kullanılmaktadır [4]. Verilerin artması ve internet teknolojilerinin gelişmesine paralel olarak makine öğrenmesi yöntemleri de gelişmekte ve günlük yaşamda kullanımı yaygınlaşmaktadır. Bu gelişmeler, istemci-sunucu sistemlerinde üretilen verilerin yönetilmesi ve verilerden çıkarım yapılması gerekliliğini doğurmaktadır. Verilerden çıkarım yapılması için istemci sunucu modellerini kullanan sitemlerde de makine öğrenmesi temelli çözümler kullanılmaktadır.

1.1. Genel Bakış

Dijitalleşen dünyada internet teknolojilerinin kullanımı her geçen gün artmaktadır. Buna paralel olarak üretilen veri miktarı da artmaktadır. Üretilen veriler, içerisinde kişisel verileri barındırabilmektedir. Tüm verilerden anlamlı sonuçlar çıkartmak için makine öğrenmesi yöntemleri kullanılmaktadır. Verileri işleyen bu yöntemler değişen ve artan veri miktarı ile gelişmektedir. Daha güncel ve doğru sonuçlar üretmektedir. Böylece doğru analizler yapmayı sağlayan makine öğrenmesi yöntemleri ekonomiden ticarete, eğitimden sağlığa kadar pek çok alanda işlevsel olarak kullanılmaktadır. Fakat kişisel verilerin kullanımı ile özel yaşamın gizliliği ve veri mahremiyeti sağlanamamakta ve bununla beraber güvenlik sorunları ortaya çıkmaktadır [5]. Tüm bu problemlere çözüm olarak ortaya çıkan federe öğrenme, veri mahremiyetinden ödün vermeden model eğitimi yapan güncel bir teknolojidir [6]. Şekil 1.2.'de federe öğrenmenin yapısı görsel olarak anlatılmaktadır [7].



Şekil 1.2. Federe Öğrenme yapısı.

Federe Öğrenme; istemcilerin (uçta çalışan cihazlar) ve sunucuların birlikte çalışması gereken sistemlerde istemcilerden sunucuya tüm veri gönderilmeden ve yeni veri geldiğinde bütün veriyi kullanmadan makine öğrenmesi modeli eğitimi için geliştirilen bir mimaridir. Mimariye göre her bir istemcide kendi verilerinden bir makine öğrenmesi modeli eğitilmektedir. Eğitilen model sunucuya gönderilmekte ve sunucuda bu modeller birleştirilerek yeni bir model oluşturulmaktadır. Oluşturulan nihai model tekrar istemcilere dağıtılmaktadır. Verinin bulunduğu alanda işlenmesi ile veri mahremiyetinden ödün verilmemiş, veri güvenliği sağlanmıştır.

1.1.1. Problemin Tanımı

İstemcilerin ve sunucuların birlikte çalışması gereken sistemlerde makine öğrenmesi modeli kullanılması bir ihtiyaçtır. Ancak istemcilerden verilerin toplanması, sunucuya aktarılması, makine öğrenmesi modeli eğitilmesi ve ardından bu modelin istemcilerde çalışan cihazlara entegre edilmesi birçok problemi beraberinde getirmektedir. Verilerin istemcilerden sunucuya transferi ağ trafiğine sebep olmakta ve fazla enerji gerektirmektedir. Bütün veriler aktarılacağı için veri mahremiyeti de istismar edilebilmektedir. Tüm bu problemlere çözüm olarak ortaya çıkan federe öğrenme teknolojisi güncel bir çalışma alanıdır. Veri mahremiyetinden ödün vermeden geliştirilecek olan makine öğrenmesi uygulamalarına yönelik talep yükselmektedir. Bu sebeplerden federe öğrenme mimarisi üzerinde yapılacak çalışmalara ihtiyaç duyulmaktadır.

1.1.2. Amaç ve Hedef

Bu çalışma ile veri mahremiyeti dikkate alınarak, istemci sunucu sistemlerinde ilişkisel sınıflandırmaya dayalı model eğitimi amaçlanmıştır. Federe öğrenme mimarisinin benimsendiği çalışmada, her bir istemcide yerel model eğitimi gerçekleştirilip modeller sunucuda birleştirilecektir. İlişkisel sınıflandırma yöntemi ile son zamanlarda her alanda kullanılmaya başlayan ve geliştirilmeye açık olan federe öğrenme mimarisi alanında çalışma yapılmamış olması, ilişkisel sınıflandırma yöntemini benimsemedeki motivasyon olmuştur. Bu mimari yaklaşımı ile ağ trafiğinin azalması, enerji ihtiyacının düşmesi ve veri mahremiyetinin bütün veriler yerine tek başına anlamsız olan veriler sunucuya gönderildiği için korunması hedeflenmektedir. Ayrıca tüm istemcilerdeki modellerin güncel olması beklenmektedir.

1.1.3. Ana Katkılar

Bu çalışma kapsamında federe öğrenme mimarisine ait Python diliyle hazırlanmış bir simülasyon ortamı oluşturulmuştur. Oluşturulan bu özgün simülasyon ortamında iki farklı şekilde model eğitimi gerçekleştirilmiş ve elde edilen iki model kıyaslanmıştır. İlk model federe öğrenme olmadan eğitilmiş modeldir. Yani istemcilerden sunucuya verilerin aktarılmasıyla ve bu verilerden model eğitilmesiyle elde edilmiştir. İkinci model ise federe öğrenme mimarisi ile, istemcilerde model eğitilip sunucuda modellerin birleştirilmesiyle oluşan nihai modeldir. Her iki model eğitiminde yaygın olarak kullanılan hazır veri setleri kullanılmıştır. Deneysel sonuçlar federe öğrenme ile eğitilen modelin diğer model ile neredeyse aynı doğruluğu elde ettiğini ama daha kısa eğitim süresine sahip olduğunu göstermiştir. Böylece federe öğrenme mimarisini kullanacak cihazlarda enerji ihtiyacının düştüğü sonucuna varılmaktadır. Ayrıca veri yerine eğitilmiş model merkeze gönderildiği için hem veri mahremiyeti sağlanmış hem de ağ trafiği kayda değer şekilde azalmıştır.

1.1.4. Tez Organizasyonu

Tez dört bölümden oluşmaktadır. “Giriş” bölümünde federe öğrenme ve ilişkisel sınıflandırma konuları anlatılmıştır. Ayrıca literatür çalışmalarına yer verilmiştir. İkinci bölüm “Yöntem” bölümüdür. Bu bölümde, kullanılan veri setleri ve federe öğrenme mimarisi ile model eğitmek için ilişkisel sınıflandırmaya dayalı geliştirilen algoritma detaylıca anlatılmıştır. Üçüncü olarak “Bulgular ve Tartışma” bölümünde deney ortamı, deney performansı ve sonuçlardan bahsedilmiştir. Son bölüm “Sonuçlar” bölümünde ise çalışmadan elde edilen bulgular, literatüre sağlanan katkılar paylaşılmıştır. Ayrıca gelecekteki çalışmalar hakkında bilgi verilmiştir.

1.2. Federe Öğrenme

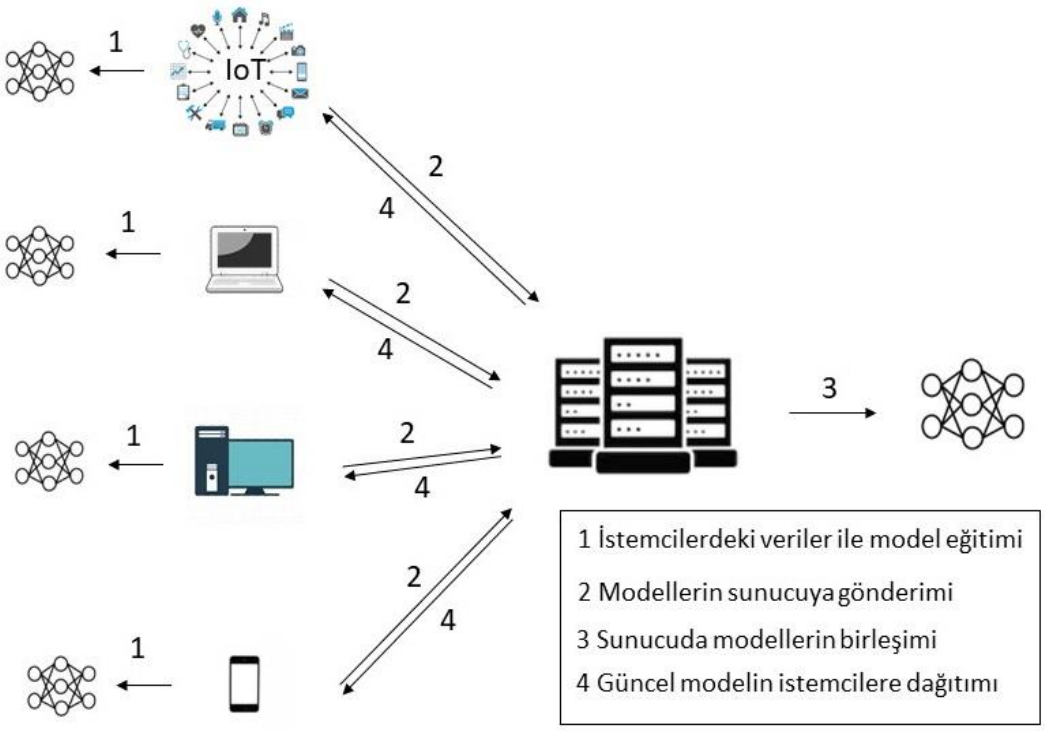
Veriler arasındaki ilişkiyi bulabilmek ve anlamlı sonuçlar çıkarabilmek için matematiksel ve istatiksel işlemler yapan algoritmalara makine öğrenmesi denilmektedir. Veri üzerinde makine öğrenmesi işlemlerinden sonra, işlenen veriden sonuçlar çıkartan bir model elde edilmektedir. Elde edilen bu model yeni veri geldikçe öğrenmeye devam etmekte ve performansını geliştirmektedir.

Makine öğrenmesi alanında geliştirilen algoritmalar belirli tipte problemleri istenen çözüme kavuşturmak için tasarlanmıştır. Problemlerin daha kolay çözüme kavuşabilmesi adına hangi tip problem olduğu belirlenmesi için kategorize edilmesi gerekmektedir [8]. Bu yüzden makine öğrenmesi teknikleri geliştirilmiştir. Bu teknikler; denetimli öğrenme [9], denetimsiz öğrenme [10], yarı denetimli öğrenme [11] ve pekiştirmeli öğrenme [12] olarak 4 ana başlık altında toplanmaktadır. Fakat bu öğrenme tipleri veriyi işlerken mahremiyet alanında eksik kalmaktadır.

Ülkemizde başta olmak üzere, makine öğrenmesi yöntemleri kullanılırken veri mahremiyeti adına oluşabilecek sorunların kanunlar ile önüne geçilmeye çalışılmıştır. Genel Veri Koruma Yönetmeliği (GDPR) ve Kişisel Verileri Koruma Kanunu (KVKK) alınan önlemlerden bazılarıdır. Kanunlar kullanıcının izni olmadan kullanıcı verilerinin işlenmesine izin vermemektedir. Fakat bu kanunlar ve yönetmelikler uygulama alanından ziyade hukuki yönü ele alındığı için veri mahremiyeti konusuna yeterli çözümü getirememektedir [13].

Veri mahremiyeti kavramı üzerine tasarlanan federe öğrenme yeni bir makine öğrenmesi tekniğidir [14]. Federe öğrenme akıllı cihazlar, mobil uygulamalar gibi istemci sunucu sistemlerinde kullanılmak üzere geliştirilmiştir. Klasik makine öğrenmesi gibi istemcilerde bulunan verileri sunucuda toplayıp, sunucuda model eğitimi yapmamaktadır. Bunun aksine verilerin bulunduğu istemcilerde yerel modeller üretip modelleri sunucuda birleştirilmektedir. Ardından sunucuda birleşen bu ana modeli tekrar istemcilere dağıtarak güncel bir model elde edilmesini sağlamaktadır. İstemciden veri yerine eğitilen model sunucuya gönderildiği için veri mahremiyeti ihlalinin önüne geçilmektedir.

1.2.1. Federe Öğrenme Mimarisinin İşleyişi



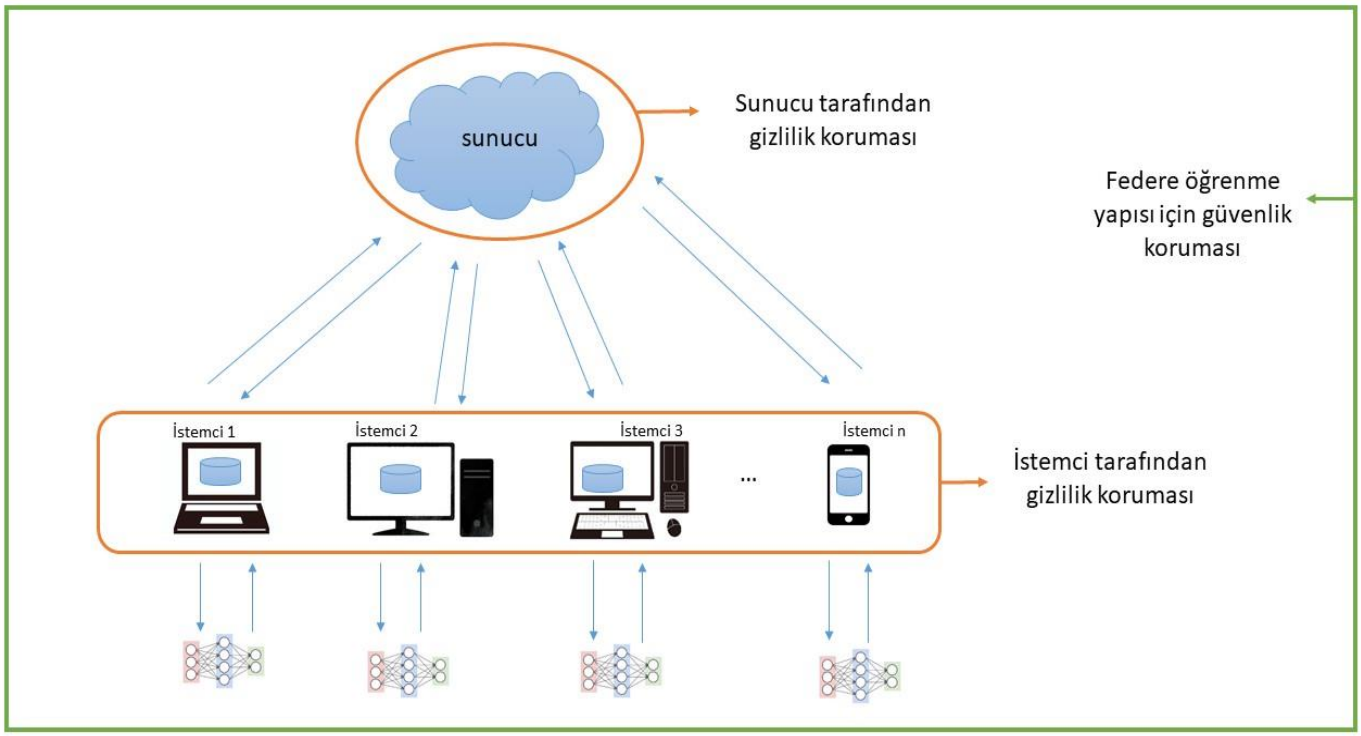
Şekil 1.3. Federe öğrenme mimarisinin işleyişi.

Şekil 1.3.'te federe öğrenmenin işleyişi ile ilgili yapıya yer verilmiştir. Şekilde ifade edildiği üzere federe öğrenme, birbirinden bağımsız ve ortak sunucuya bağlı iki veya daha fazla cihazda makine öğrenmesi yöntemini veri mahremiyetinden ödün vermeden gerçekleştirmektedir. Model eğitimi yerelde yani istemcilerde kendi verileri ile gerçekleşmektedir. Daha sonra eğitilen modeller sunucuya gönderilip burada toplanarak birleştirilmektedir. Oluşan nihai model uçlara dağıtılmaktadır. Böylece her istemcide hem güncel modeller olmaktadır hem de yerelden veri çıkmadan daha fazla veri ile eğitilmiş modeller kullanılmaktadır.

Veri mahremiyeti ve kişisel verilerin güvenliği adına oluşan problemlere çözüm olarak, ilk defa Google 2016 yılında federe öğrenme kavramını ortaya koymuştur. Veri gizliliği temelli oluşturulan bu öğrenme modeli ile finansal, tıbbi ve kentsel alanda veri güvenliği yeniden oluşturulmuştur [15].

İstemci sunucu mimarisi ile çalışan yapılarda makine öğrenmesi modeli kullanılması ihtiyaç haline gelmiştir. Fakat bu yapılar için sunucuya verilerin gönderilip burada model eğitimi yapılması veri mahremiyeti ve veri gizliliği problemini ortaya çıkartmaktadır. Federe öğrenme, istemci sunucu mimarisinde ortaya çıkan bu problem için geliştirilmiştir [16]. Ayrıca veri mahremiyetini sağlaması dışında federe öğrenme ile istemci sunucu mimarisini kullanan sistemlerde, iletişim maliyeti azalmaktadır. Pille çalışan cihazlar için güç tasarrufu gerçekleşmektedir. Bağlantı kurma ile ilgili gecikme ortadan kalkmaktadır [6].

1.2.2. Federe Öğrenme Mimarisinde Veri Gizliliği ve Veri Güvenliği



Şekil 1.4. Federe öğrenme mimarisinde veri gizliliği ve güvenliği.

Federe öğrenme yaklaşımında veri mahremiyetini sağlamak asıl amaçtır. Bunun için veri gizliliği ve güvenliği üzerine çalışmalar yapılmaktadır [17].

Şekil 1.4' te anlatıldığı üzere veri gizliliği istemci ve sunucu alanlarında sağlanmaktadır. Veri güvenliği ise federe öğrenme mimarisinde korunmaktadır.

İstemci tarafında iki yöntem ile veri gizliliği sağlanmaktadır.

1. Bozulma (Perturbation)

İstemciden sunucuya gönderilen parametrelere gürültü ekleyerek veri gizliliği sağlanmaktadır.

2. Kukla (Dummy)

İstemcide eğitilen modellerde bulunan parametrelere sahte parametreler ekleyip sunucuya göndererek veri gizliliği sağlanmaktadır.

İstemci tarafında alınan bu önlemler ile veriler işlenemez hale getirilerek saldırganlar tarafından kişisel bilgilere ulaşım engellenmektedir.

Federe öğrenme mimarisine göre, tüm istemcilerden modeller ve modeller ile gelen parametreler sunucuda toplanmaktadır. Ardından burada nihai model elde edilmekte ve istemcilere tekrar dağıtılmaktadır. Dağıtım sırasında saldırganlarca veri gizliliği ihlalinin önüne geçmek için sunucu tarafında iki yöntem sunulmaktadır.

1. Toplama/Birleştirme (Aggregation)

Bu yöntem ile her bir istemcideki kullanıcı bilgileri daha karmaşık hale getirilip saldırganlar tarafından anlaşılırlığının önüne geçilmesi için, farklı kullanıcıların parametreleri yan yana getirilip işlem yapılmaktadır.

2. Güvenli Çok Taraflı Hesaplama (Secure Multi-Party Computation, SMC)

SMC, İstemciden sunucuya ve ya sunucudan istemciye gönderilen parametreleri şifreleyerek gönderim sağlanmaktadır. Böylece veri gizliliği sağlanmaktadır.

Federe öğrenme mimarisi kapsamında veri güvenliğini sağlamak adına ise iki yöntem önerilmektedir.

1. Arka Kapı Saldırısı (Back-door Defender)

İstemciler içerisinde kötü niyetli saldırganlar bulunabilir. Bunu federe öğrenme mimarisi bilemediği için istemciler tarafından modelin güncellenmesine izin verilmemekte ve istemciler denetlenmektedir.

2. Homomorfik Şifreleme (Homomorphic Encryption)

Makine öğrenimi süresi boyunca model parametreleri şifrelenmektedir.

1.2.3. Federe Öğrenme Modelleri

Veri seti içinde satırlar verilerin toplandığı varlığı, sütunlar ise öznitelikleri ifade etmektedir. Veri paylaşımına dair problemlerin çözümü için 3 çeşit federe öğrenme modeli mevcuttur.

- Yatay Federe Öğrenme Modeli

Her uç cihazda üretilen verilerin öz niteliklerinin aynı fakat varlıkların farklı olduğu durumlarda yatay federe öğrenme modeli kullanılmaktadır.

- Dikey Federe Öğrenme Modeli

Her uçtan gelen veriler içerisindeki öznitelikler az sayıda örtüşmektedir. Yani model eğitimi için varlıklar aynıdır ama öz nitelikler kısmen ortaktır. Bu gibi durumlarda dikey federe öğrenme modeli kullanılmaktadır.

- Federe Transfer Öğrenme Modeli

Her uç cihazda üretilen verilerin hem öz niteliklerin hem de varlıkların farklı olduğu durumlarda federe transfer öğrenme modeli kullanılmaktadır.

1.2.4. Federe Öğrenme Kullanım Alanları

Günümüzde federe öğrenme birçok alanda kullanılmaktadır ve geliştirilmeye açık bir teknolojidir. Sağlık, eğitim, akıllı şehir, giyilebilir cihazlar, finans, blok zincir ve nesnelerin interneti gibi önemli ve geniş bir alanda uygulamalara sahiptir [18, 19]. Tablo 1.1'de federe öğrenme mimarisinin uygulandığı alanlar ve o alanlara ait yapılan çalışmalar yer almaktadır.

Tablo 1.1. Federe öğrenme mimarisinin uygulandığı alanlar.

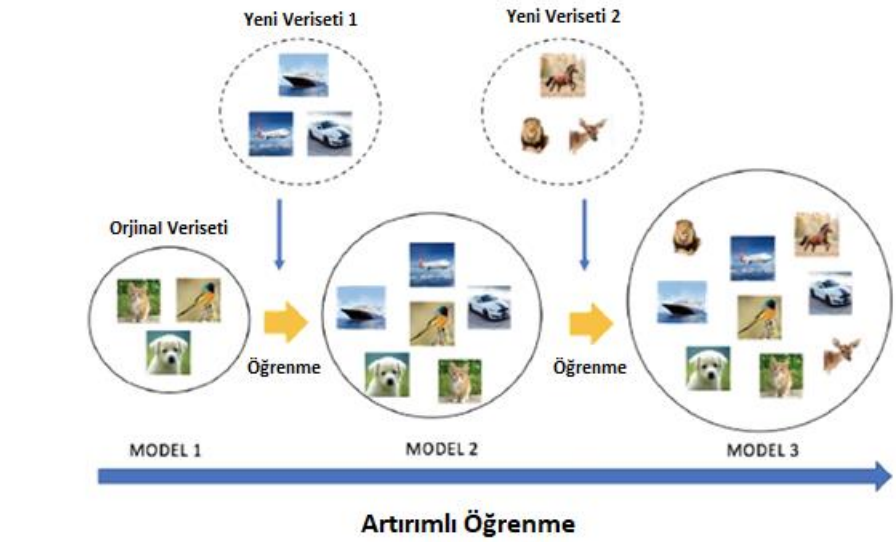
Uygulama Alanı	Yapılan Çalışmalar
Nesnelerin İnterneti	Ağ anormalliği algılama Ağ saldırı algılama Ağdaki kullanıcı ilişkilendirmesinin tahmini Sanal gerçeklik Arttırılmış gerçeklik Yapay zekâ ile nesne algılama Elektrikli araç pillerinin arızalanması ve arıza tespiti Mobil cihazlara dayalı anahtar kelime tespiti Mobil cihazlarda klavye tahmini Mobil cihazlarda klavye emoji tahmini Sözlük dışı kelimeleri öğrenme Sensör verilerine dayalı insan hareketlerini tanıma Cihaz verilerine dayalı insan davranışı ve duygularında elektroensefalografi sınıflandırması Mobil cihazlara dayalı insan hareketliliği tahmini Web tarayıcı deneyim kalitesi tahmini Firefox uygulaması arama motoru geliştiricisi Mobil cihaz bilgi takibi uç cihazların anormallik tespiti Görsel soru cevaplama Yapay Zekâ ve Büyük Veri Çalışmaları Siber Güvenlik ve Mahremiyet
Sağlık	Erken doğum tahmini Erken gebelikte kilo alımı tahmini Klinik verilere dayalı tanı konulması Hastalık tahmini Beyin tümörü segmentasyonu Fonksiyonel manyetik rezonans görüntüleme analizi İlaçla ilgili özelliklerin tahmini Ortak ilaç gelişimi Ağrı yüz ifadesi tanıma Giyilebilir sağlık cihazları hizmetleri Klinik doğal dil işleme

Kent Bilgi Sistemleri ve Akıllı Şehir	Araç kamerası ile toplanan görüntülerin sınıflandırması ve etiketlemesi Hava kalitesi değerlendirmesi Elektrikli araç ağı için enerji talebi tahmini Trafik işareti tanıma Trafik akışı tahmini Araç planlaması Araç yönlendirme Güvenlik izleme Enerji talep tahmini
Fiziksel Bilgi Sistemleri	Yer eşleştirme Otonom sürüş Bulut robotik sistemler Drone yayını bozma saldırısı algılama Havacılıkta başarısızlık tahmini
İnternet ve Finans	Kredi kartı dolandırıcılık tespiti Mali suçların tespiti Finansal metin tanıma Kişiselleştirilmiş arama Haber önerisi
Endüstri	Endüstriyel konu modelleme Ortam durumu izleme Ürün görsel incelemesi Sensör arızası tahmini

1.3. Artırımlı Öğrenme

Makine öğrenmesi ile oluşturulacak modelin öğrenme işlemi, tek seferde öğrenme (Batch learning) şeklinde gerçekleşir. Fakat bu öğrenme şeklinde eğitilen model, sürekli değişen taleplere cevap verememektedir. Değişen ve sürekli gelen veri kümelerine uyum sağlayarak telepleri karşılayan güncel modeller, sürekli/artırımlı öğrenme (Incremental learning) ile sağlanmaktadır [20]. Artırımlı öğrenmenin amacı, eğitilen modelin doğruluğundan ödün vermeden yeni gelen veriler ile öğrenmeye devam etmesidir. Artımlı öğrenmenin tek seferde öğrenmeden farkı, modelin gelen her yeni bilgi ile kendini güncellemesidir [21].

Bir örnek üzerinde Türkiye'de market sepeti analizi anlatılmak istenirse, kolonya ürünü dini bayramlarda çikolata, şeker, lokum vb. ürünler ile birlikte satılırken, bu günlerde pandemiden dolayı bu ürün eldiven, maske, dezenfektan ürünleri ile birlikte satılmaktadır. Müşterilerin alışkanlıkları zaman içerisinde değişebilmektedir. Dolayısıyla birlikte aldıkları ürünlerde değişiklik arz edebilmektedir. Artırımlı öğrenme ile model eğitildiğinde değişen alışkanlıklara adaptasyon sağlanabilmektedir.



Şekil 1.5. Artırımlı öğrenme işleyişi.

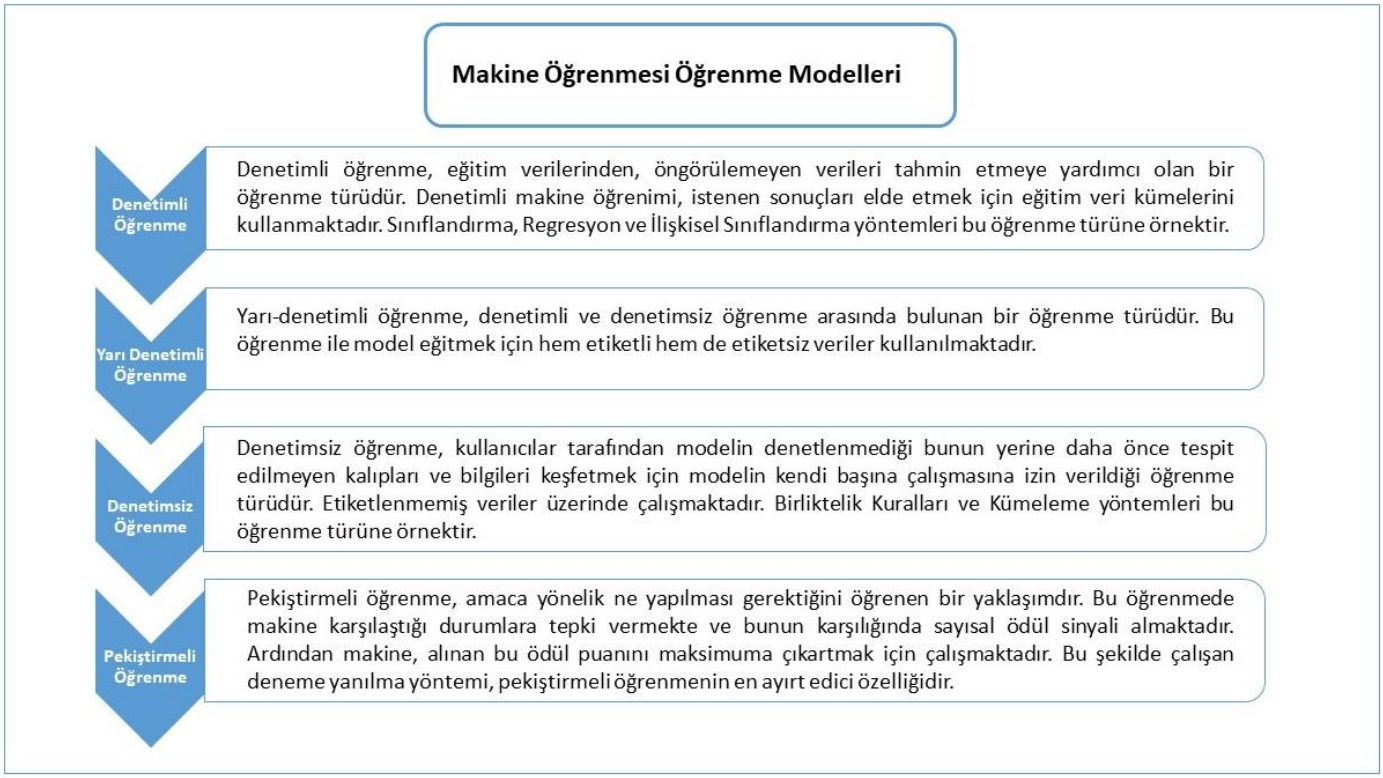
Şekil 1.5 'e göre model zaman içerisinde verilerdeki değişikliklere adapte olabilmek için gelen güncel veriler ile öğrenmeye devam etmektedir. Model eğitiminde kullanılacak olan yöntem ile yeni gelen veriden tekrar öğrenme gerçekleşmektedir. Bellek sınırı aşılmadan ve olabildiğince hızlı bir şekilde model güncellenir. Ardından bir sonraki örnek için hazır hale gelmiş olmaktadır. Bu her yeni gelen veri için tekrar etmektedir [22].

1.4. İlişkisel Sınıflandırma

Teknolojinin gelişmesi ile birlikte kullanılan uygulamalar ve cihazlar vasıtasıyla insanlar günlük yaşamlarının her alanında veri üretir duruma gelmiştir. Gün geçtikçe üretilen veri miktarının artması sonucunda verilerin işlenerek anlamlı sonuçların üretilmesi veri madenciliği alanını ortaya çıkartmıştır. Bu bağlamda veri madenciliği, mevcut büyük veriler üzerinde bir takım işlemler yaparak önceden anlamsız olan verilerden geçerli ve kullnılabilir bilgileri ortaya çıkartan bilim dalı olarak tanımlanmaktadır. Veriler üzerinde çıkarım işlemi için birçok farklı makine öğrenmesi metodu kullanılmaktadır. Bunlar sınıflandırma (classification), kümeleme

(clustering), regresyon (regression), birliktelik kuralları (association rules) ve ilişkisel sınıflandırma (associative classification) olarak adlandırılmaktadır.

Sınıflandırma, regresyon ve ilişkisel sınıflandırma yöntemleri eğitim verisine ihtiyaç duyduğu için denetimli öğrenme başlığı altında yer almaktadır. Birliktelik kuralları ve kümeleme yönteminde ise eğitim verisi gerekmediği için denetimsiz öğrenme olarak adlandırılır. İnsan gözetiminin olup olmama durumuna göre ayrılmış makine öğrenmesi yöntemleri detaylı bir şekilde Şekil 1.6.'da görsel olarak ifade edilmektedir.



Şekil 1.6. Makine öğrenmesi yöntemleri için kullanılan öğrenme tipleri.

Etiketli verilerden tahminde bulunarak çıktı üretmek için sınıflandırma ve regresyon yöntemleri kullanılmaktadır. Eğer çıktı kategorik ise sınıflandırma, nümerik ise regresyon ile tanımlanır. Etiketli olmayan veri seti için, veriler arasında benzer özellikler gösterenleri aynı gruplara ayırma işlemi, kümeleme yöntemi ile gerçekleştirilmektedir. Aynı küme içinde benzerlikler fazla, kümeler arası benzerlikler azdır. Birliktelik kuralları, veriler arasında bağlantılar bulup kurallar çıkartan bir yöntemdir.

İlişkisel sınıflandırma ise, sınıflandırma ile birliktelik kurallarını birleştiren veri madenciliği yöntemlerinden biridir [23]. İlişkisel sınıflandırma yönteminin daha iyi anlaşılabilmesi için bu başlık altında sınıflandırma ve birliktelik kuralları yöntemleri anlatılmaktadır. Ardından ilişksel sınıflandırma konusu detaylandırılacaktır.

1.4.1. Sınıflandırma

Sınıflandırma, model oluşturmak için kullanılan veriden elde ettiği tecrübe ile yeni gelen verileri tahmin etme işlemidir. Sınıflandırma yöntemi ile model oluşturmak için önce işlenecek veri seti, eğitim ve test veri seti olarak ayrılmaktadır. Veri seti içerisinde verinin özellikleri ve bu özelliklere ait sonuç bilgisi yer almaktadır. Eğitim verisi üzerine sınıflandırma yöntemlerinin birisi uygulanmaktadır. Uygulama sonrasında bir model elde edilmektedir. Elde edilen bu modelin test veri seti ile doğruluğu denenmektedir. Daha sonra yeni veriler üzerinde model tahmin ederek sınıflandırma işlemini gerçekleşmektedir. Model tahmin sonucunda çıktı olarak kategorik bilgiler üretmektedir.

Makine öğrenmesi uygulamalarında sınıflandırma yapan yöntemlerden bazıları aşağıda listelenmiştir:

- En Yakın Komşu Algoritması (K-NN)
- ZeroR
- OneR
- Naive Bayes Sınıflandırıcısı
- Karar Ağaçları (Decision Trees)
- Lineer Diskriminant Analizi
- Bayes Ağları
- Çok Katmanlı Perceptrons (Multi-Layer Perceptrons)
- Destek Karar Makineleri (Support Decision Machines)
- Yapay Sinir Ağı (Artificial Neural Network)

1.4.2. Birliktelik Kuralları

Birliktelik kuralları analizi veri madenciliğinde kullanılan ilk tekniklerden birisidir. Aynı zamanda veri madenciliği alanında üzerinde çok fazla araştırma ve çalışma yapılmış olan ilgi çekici bir konudur.

Birliktelik kuralı, veri seti içerisinde bir birleri ile ilişkili olan özelliklerin ortaya çıkmasını sağlayan bir yöntemdir. İlişkili veriler ile birliktelik davranışlarının tespit edilmesi sayesinde geleceğe yönelik çıkarımlar yapılabilmektedir.

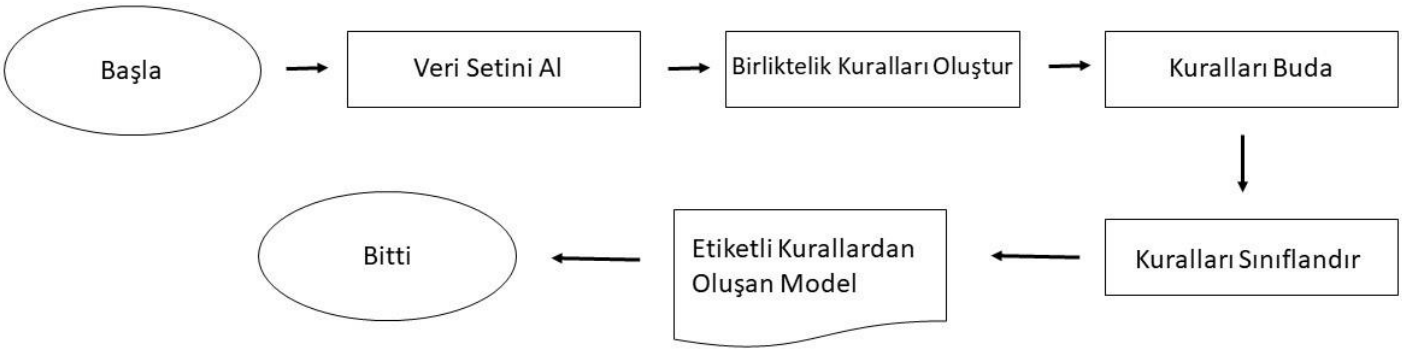
Birliktelik kurallarının kullanıldığı en tipik örnek market sepeti uygulamasıdır. Uygulamaya göre yöntemin amacı alışveriş esnasında müşterilerin satın aldıkları ürünler arasındaki birliktelik ilişkisini bulmak, bu ilişki verisi doğrultusunda müşterilerin satın alma alışkanlıklarını tespit etmektir. Satıcılar, keşfedilen bu birliktelik bağıntıları ve müşteri alışkanlıkları sayesinde kazançlı pazarlama ve satış imkanına sahip olmaktadırlar.

Yöntem uygulanırken kullanılacak veri setinden, matematiksel olarak $X \Rightarrow Y$ şeklinde ifade edilen kurallar oluşturulur. $X \Rightarrow Y$, X birliktelik Y şeklinde okunmaktadır. X kuralın sol tarafını (lhs: Left hand side), Y ise kuralın sağ tarafını (rhs: Right hand side) göstermektedir. X ve Y veri seti içinde bir yada birden fazla ilişki bulunan verilerdir. Kuralları oluşturabilmek için destek (support) ve güven (confidence) değerlerini kullanılmaktadır. Kullanıcı tarafından belirlenmiş minimum destek ve minimum güven değerleri ile yaygın birlikteliklerin belirlenmesi amaçlanmaktadır. Veriler arasındaki ilişkinin büyüklüğü elde edilen kuralların güçlülüğünü göstermektedir.

Birliktelik kuralları için kullanılan algoritmalar aşağıdaki gibidir:

- AIS
- SETM
- OCD
- CARMA
- Partitioning
- Sayım Dağılımı (Count Distribution)
- Akıllı Veri Dağıtımı (Intelligent Data Distribution)
- Paralel Birliktelik Kuralları (Parallel Association Rules)
- Apriori-Hybrid
- AprioriTid
- Apriori

Bu algoritmalar arasında en çok kullanılan ilişkilendirme algoritması Apriori'dir.



Şekil 1.7. İlişkisel sınıflandırma yöntemi çalışma adımları.

Şekil 1.7.'de ilişkisel sınıflandırma yöntemi ile model eğitimi için uygulanan adımlar yer almaktadır. Şekle göre bu yöntem ile önce birliktelik kurallarına dayalı kurallar oluşturulur. Daha sonra oluşturulan bu kurallar sınıflandırılarak denetimli öğrenme modeli kurulur.

Yöntem uygulanırken kullanılacak veri setinden $X \Rightarrow Y$ şeklinde kurallar oluşturulur. İlişkisel sınıflandırma yöntemi ile oluşturulan kurallarda kuralın sol tarafı olan X, birliktelik kuralları yöntemindeki gibi, veri setinde birlikte bulunan örnekleri göstermektedir. Fakat kuralın sağ tarafı olan Y, sadece sınıf değişkenini ifade etmektedir [24]. Bu noktada ilişkisel sınıflandırma ile birliktelik kuralları yöntemi birbirinden ayrılmaktadır.

İlişkisel sınıflandırma yönteminin çalışması üç adımda gerçekleşmektedir. İlk adım oluşturulacak kurallar için en küçük destek ve güven değerlernin belirlenmesidir. İkinci adım veriler ile kuralların oluşturulup zayıf kuralların budanma işlemidir. Üçüncü adım olan son adımda ise en iyi sınıflandırma yapan kuralların belirlenmesi aşamasıdır [25].

İlişkisel sınıflandırmada kullanılan birçok algoritma bulunmaktadır [24]. Bu amaçla kullanılan algoritmaların ilki, classification based on association (CBA) algoritmasıdır [23].

Literatürdeki çalışmalar birliktelik kuralları ile sınıflandırmayı birleştiren ilişkisel sınıflandırma yöntemiyle eğitilen modelin karar ağaçları ile eğitilen modelden doğruluk değeri daha yüksek sonuçlar ürettiğini göstermektedir [23, 26, 27]. Bu yöntem ile "eğer -> sonuç" şeklinde son kullanıcı tarafından yorumlanması kolay kurallar üretildiği için bir çok alanda kullanılmaktadır [28, 29].

1.5. Literatür Taraması

İlişkisel sınıflandırma yöntemi ve federe öğrenme mimarisine ilişkin literatürde birçok çalışma bulunmaktadır. Bu bölümde, bahsedilen alanlara yönelik yapılan çalışmalar özetlenmektedir.

İlişkisel sınıflandırma birliktelik kurallarına dayalı olarak sınıflandırma yapan veri madenciliği yöntemidir [23]. Yöntem ile oluşturulan kurallar etiketlenerek sınıflandırılır ve kullanıcı tarafından daha kolay yorumlanır. Bu yöntem ile birlikte, yeni bir algoritma da ortaya çıkmıştır. Algoritma, birliktelik kurallarına dayalı sınıflandırma (classification based on association rules, CBA) olarak adlandırılmaktadır. Algoritmanın çalışma mantığı üç aşamadan oluşmaktadır: Veriler ile kural oluşturma ilk aşamadır. Zayıf kuralların budanma işlemi ikinci aşamadır. Son aşama ise en iyi sınıflandırma yapan kuralların elde edilmesi aşamasıdır. Çoklu sınıf ilişkilendirme kurallarına dayalı sınıflandırma (Classification based on multiple class-association rules, CMAR) [30] ve birliktelik kuralına dayalı çok sınıflı sınıflandırma (multiclass classification based on association rule, MCAR) [26] adı verilen algoritmalar da ilişkisel sınıflandırma yöntemi için geliştirilen algoritmalardır. Bahsedilen bu algoritmaların çalışma mantığı ortaktır ancak kural oluşturma süreçleri birbirinden farklıdır. Bir diğer ortak özellikleri de geleneksel sınıflandırma yöntemlerinden biri olan karar ağaçlarından daha doğru sınıflandırma başarısına sahip olmalarıdır.

İlişkisel sınıflandırma metodunda kullanılan CBA algoritması üzerinde, modelin doğruluk değerini ve eğitim süresini arttıran iyileştirmeler yapılarak gelişmiş CBA (enhanced CBA, ECBA) [31], birliktelik kuralına göre hızlı sınıflandırma (fast classification based on association rule, FCBA) [32] ve birliktelik kurallarına dayalı ağırlıklı sınıflandırma (weighted classification based on association rules, WCBA) [33] isimli algoritmalar geliştirilmiştir. Fakat bu algoritmalar ile geliştirilen modeller güncelliğini koruyamamaktadır. Sürekli değişen talepler neticesinde veriler değişime uğrar. Eğitilen modellerin de doğru ve hızlı çalışmasının yanı sıra bu değişime uyum sağlaması beklenir. Değişen veri kümelerine modelin uyum sağlayarak güncel kalması artırımlı öğrenme yöntemi (incremental learning) ile sağlanmaktadır [20]. CBA algoritması geliştirilerek modelin güncel kalmasını sağlayan artırımlı madenciliğe dayalı ilişkili sınıflandırma (associative classification based on incremental mining, ACIM) [34] ve artırımlı madencilik algoritmasına dayalı gelişmiş ilişkisel sınıflandırma (enhanced associative classification based on incremental mining algorithm, E-ACIM) [35] algoritmaları ilişkisel sınıflandırmada artırımlı öğrenme yöntemine örnek verilebilir.

Gboard mobil kullanıcılar için Google tarafından geliştirilmiş olan sanal klavye uygulamasıdır. Bu uygulama bir sonraki kelime tahmini, kelimeyi düzeltme ve tamamlama gibi özellikleri ile kullanıcılarına kolaylık sağlamaktadır. Bu özellikleri makine öğrenmesi yöntemleri ile gerçekleştirmektedir. 1,4 milyon parametreye sahip modelin eğitiminde federe öğrenme teknolojisi kullanılmaktadır. Her bir istemcide model eğitimi 5 gün sürmektedir. Eğitim ardından modeller sunucuda birleştirilmekte ve her bir istemciye birkaç dakika içinde birleştirilmiş model gönderilmektedir. Kelime tahmini, kelimeyi düzeltme ve tamamlama gibi işlemler mobil kullanıcıların kişisel verileri paylaşılmadan gerçekleştirildiği için, federe öğrenme mimarisi ile geliştirilmiş ilk uygulamalardan biri olan Gboard uygulamasının faydalı olduğu sonucuna ulaşılmıştır [7, 36, 37].

Firefox'un adres çubuğunda arama yapıldığında öneriler sunmak için yaklaşık 360.000 kişiden veri toplanmıştır. Veriler ile modelin eğitimi federe öğrenme mimarisi kullanılarak gerçekleştirilmiştir. Model internet kullanıcısının yer işaretleri, arama geçmişi gibi alışkanlıklarına bakarak öneri sunmaktadır [38].

Dolandırıcılık Tespit Sistemleri dijital bankacılıktaki şüpheli hareketleri izleyen bir yazılımdır. Jansson ve ark. bu alanda federe öğrenme mimarisi kullanarak kurumlar arası veri transferi yapmadan model eğitimi gerçekleştirmişlerdir. Böylece hem kullanıcı verileri güvenli hale getirilmiştir hem de kredi kartı dolandırıcılığının önüne geçilmiştir [39].

Nesnelerin interneti ile akıllı cihazların kullanımı sağlık alanında artmış durumdadır. Yuan ve ark. yaptıkları çalışmada, sağlık alanında kullanılan giyilebilir akıllı cihazlar için federe öğrenme ile model eğitmenin veri paylaşılmayarak hasta mahremiyetini koruduğu ve cihazlar arası iletişim ile enerji yükünü azalttığı için daha faydalı olacağını göstermişlerdir [40].

2021 yılında Kumar ve arkadaşları günlük hayatı büyük ölçüde etkileyen Covid-19 salgınına yakalanan hastaları teşhis etmek üzere çalışma yapmışlardır. Çalışma kapsamında bilgisayarlı tomografi görüntülerine bakılarak hastalık teşhisinin yapılabilmesi için mimari önerilmiştir. Geliştirilen mimari blok zincir teknolojisi ve federe öğrenme tekniğini kullanarak farklı hastaneler arasında veri mahremiyetinden ödün vermeden model paylaşmayı desteklemektedir. Model, derin öğrenme algoritmaları ile geliştirilmiştir. Geliştirilen model literatürde yapılan benzer çalışmalar ile kıyaslanmıştır. Deneysel sonuçlar geliştirilen modelin %98,68 doğruluk oranı ile çalıştığını ve benzer çalışmalara göre daha başarılı sonuçlar ürettiğini göstermektedir.

Hastaneler özel verilerini paylaştıkça model akciğer taraması kullanan COVID-19 hastalarının tespit edilmesine yardımcı olacaktır. Böylece daha küresel ve iyi bir model elde edilecektir [41].

Federe öğrenme veri mahremiyeti odaklı çalışmalarda sıklıkla ele alınmaktadır. Literatürde araştırmacılara bu alanda yardımcı olmak için inceleme çalışmaları yer almaktadır. Yu ve ark. veri madenciliğinde federe öğrenme teknolojisinin yerini anlatan çalışma paylaşmışlardır [42]. Çalışma kapsamında ilgili kavramların kapsamlı tanımlamaları yapıldıktan sonra federe öğrenmenin gelişimi ve uygulamadaki karşılaşılan zorlukları anlatılmaktadır. Sonuç olarak federe öğrenmemenin uygulama alanları detaylandırılmış ve bu alanlarda yapılan çalışmalar avantajları-dezavantajları karşılaştırılmıştır.

Literatürdeki çalışmalara bakıldığında veri mahremiyetine dayalı federe öğrenmenin birçok alanda kullanıldığı ve başarılı sonuçlar ürettiği görülmektedir [43]. Güncel ve geliştirilmeye açık olan bu çalışma konusuna araştırmacıların daha çok yönelmesi gerektiği sonucu çıkarılmaktadır. Bu çalışma kapsamında istemci-sunucu sistemleri için federe öğrenme mimarisi ile geliştirilen algoritma diğer araştırmacılara yol gösterecektir.

2. YÖNTEM

Bu tez kapsamında federe öğrenme mimarisi ile model eğitimi gerçekleştirmek için algoritma geliştirilmiştir. Geliştirilen algoritma kullanılarak eğitilen modelin test edilebilmesi için simülasyon ortamı oluşturulmuştur. Model eğitiminde güncel ve literatürde yaygın olarak çalışılan veri setleri kullanılmıştır. Çalışmanın bu bölümünde kullanılan veri setleri ve geliştirilen algoritma hakkında detaylı bilgilere yer verilmektedir.

2.1. Veri Seti

Bu çalışmada hazır veri setleri kullanılmıştır. UCI veri havuzundan car evaluation [44], bank marketing [45], mushroom [46], nursery [47] ve adult [48] veri setleri seçilmiştir. Bu veri setleri literatürde yapılan çalışmalarda güncel olarak kullanılmaktadır. du-CBA algortiması ile kullanılacak veri setinden daha anlamlı birliktelik kuralları elde edebilmek ve oluşacak modelin etkinliğini arttırmak için kategorik veriler gerekmektedir. Çalışma kapsamında kullanılan veri setleri kategorik verilerden oluştuğu ve veri seti içinde eksik (Null) veri bulunmadığı için tercih edilmiştir. Tercih edilen bu veri setleri farklı öznitelik sayısına, farklı kayıt sayısına ve farklı sınıf sayısına sahiptir. Tez kapsamında geliştirilen algoritma farklı çeşitlikteki bu veri setleri ile çalıştırılmış ve performansı gözlemlenmiştir. Tablo 2.1.'de bu veri setlerinin özellikleri yer almaktadır.

Tablo 2.1. Çalışma kapsamında kullanılan UCI veri setlerinin özellikleri.

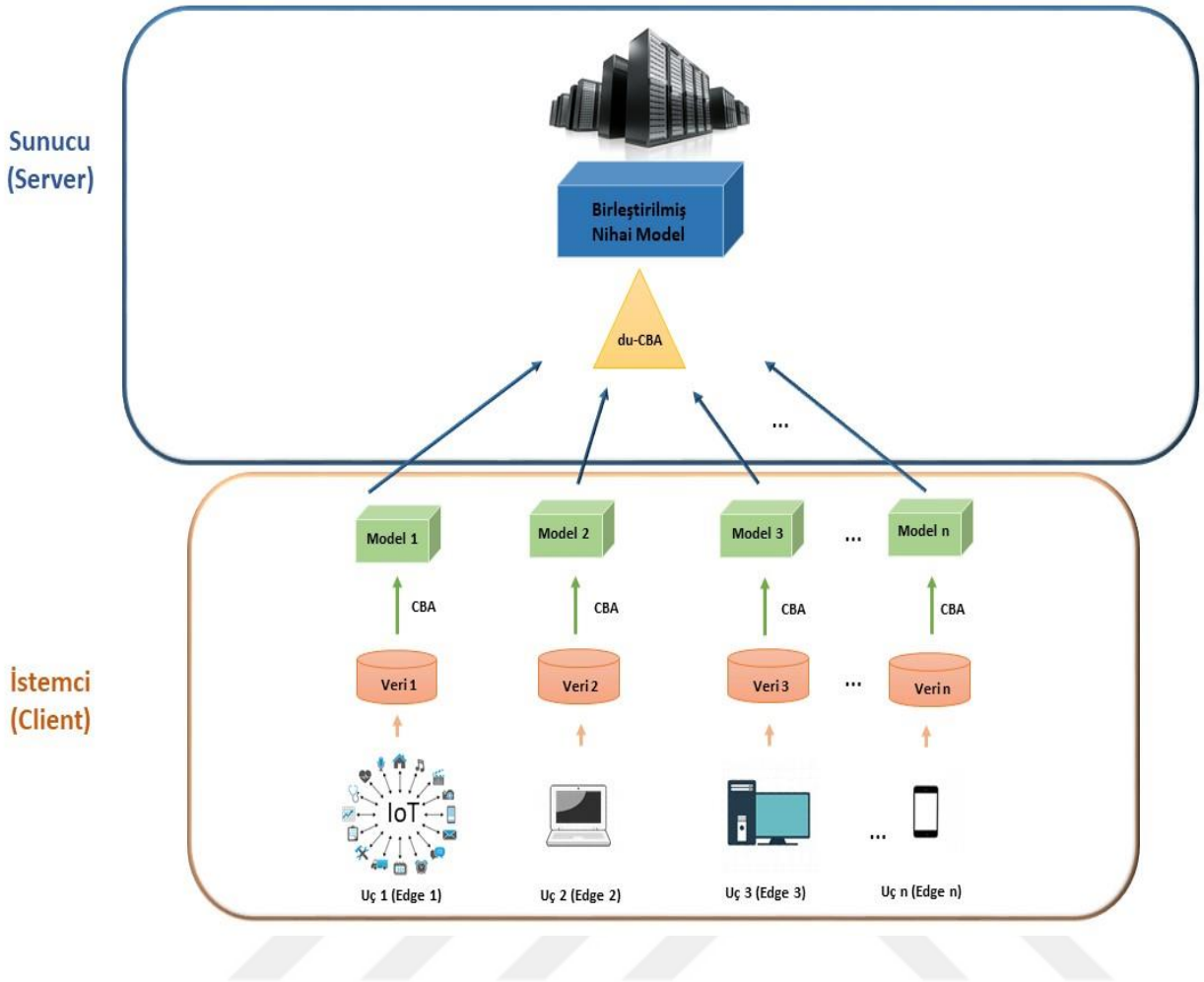
Veri Seti	Öznitelik Sayısı	Kayıt Sayısı	Sınıf
Car Evaluation	6	1728	4
Bank Marketing	16	4521	2
Mushroom	22	8124	2
Nursery	8	12960	4
Adult	14	32561	2

2.2. du-CBA

Çalışma kapsamında birliktelik kuralları ile sınıflandırma yapan du-CBA isimli ilişkisel sınıflandırma algoritması geliştirilmiştir. Geliştirilen algoritma federe öğrenme mimarisi ile model eğitmeyi amaçlamaktadır. Model eğitimini ilişkisel sınıflandırma algoritmalarından biri olan CBA algoritmasına dayalı gerçekleştirmektedir. Ayrıca geliştirilen algoritma güncel modeller eğittiği için artımlı öğrenmeyi de kolaylaştırmaktadır.

2.2.1. du-CBA ile Federe Öğrenme Mimarisi

Federe öğrenme mimarisi ile model eğitmek için geliştirilen algoritmanın mimaride kullanımı Şekil 2.1.'de sunulmaktadır. Şekil 2.1' de uçta çalışan cihazlar bir mobil cihaz, bilgisayar ya da IoT cihazlar olarak temsil edilebilir. Bu cihazlar birbirinden farklı ya da birbiri ile aynı ve farklı lokasyonlarda olabilir. Ancak cihazlarda aynı uygulama çalıştırılmakta ve eğitilen modeller her cihaz için aynı parametreler ve değişkenler ile olmaktadır. Uçlarda bulunan bu cihazlar üzerinde çalışan uygulama içerisinde, yerel veriler ile du-CBA algoritması kullanılarak modeller eğitilmektedir. Yerel veriler ile eğitilen modeller kaydedilmekte ve ardından uygulamaların ulaşabildikleri ortak bir sunucuya gönderilmektedir. Her bir istemciden eş zamanlı sunucuya gelen modeller, sunucuda çalışan uygulama içerisinde du-CBA algoritması kullanılarak birleştirilmektedir. Modellerin birleştirilmesi ile sunucuda nihai model oluşturulmaktadır.



Şekil 2.1. Federe öğrenme mimarisi ile model eğitiminde du-CBA kullanımı.

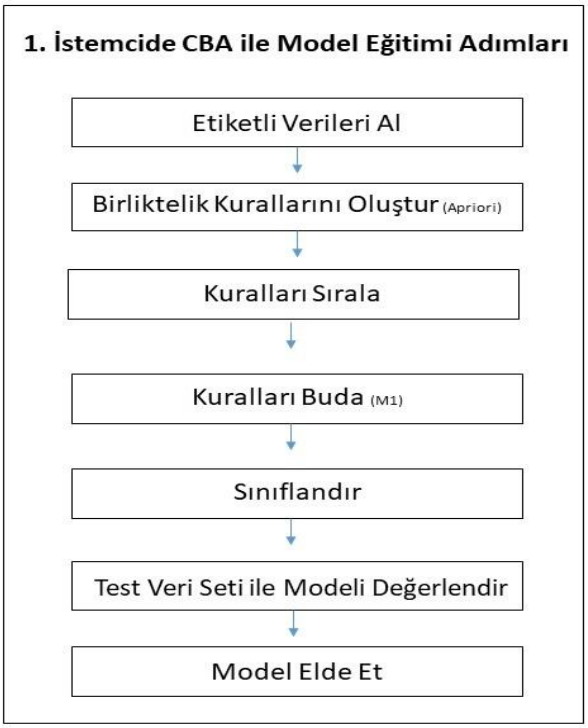
2.2.2. du-CBA Algoritması Çalışma Mantığı

Geliştirilen algoritma du-CBA, federe öğrenme mimarisi kapsamında geliştirilmiştir ve birliktelik kurallarına dayalı olarak sınıflandırma yapmaktadır.

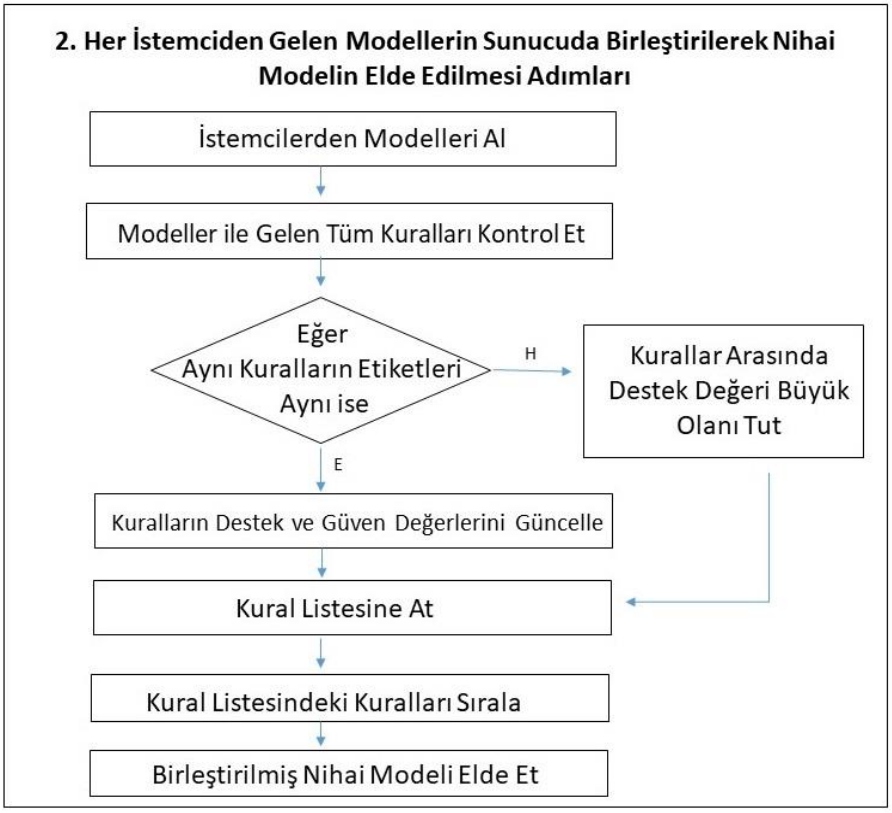
Şekil 2.2.'de du-CBA algoritmasının çalışması görsel olarak ifade edilmiştir. Şekil ardından algoritmanın ayrıntılı açıklamasına yer verilmiştir.

Şekil 2.2.'de gösterildiği üzere du-CBA algoritması iki adımda çalışmaktadır.

1. İstemcilerde/uçlarda modellerin eğitimi
2. Modellerin birleştirilmesi



(a)



(b)

Şekil 2.2. (a) du-CBA algoritmasının istemcide çalışma adımları; (b) du-CBA algoritmasının sunucuda çalışma adımları.

Birinci adımda uçtaki cihazlarda model oluşturmak için CBA algoritması kullanılmaktadır. Denetimli veri madenciliği yöntemlerinden biri olan CBA, etiketli birliktelik kuralları oluşturur (CARs). Birliktelik kurallarının oluşturulması aşaması Classification Based on Association Rule Generation (CBA-RG) olarak adlandırılır. Kuralların etiketlendiği sınıflandırma aşaması ise Classification Based on Association Classifier Building (CBA-CB) olarak adlandırılmaktadır.

CBA-RG aşamasında veriler arasındaki ilişkiler çıkartılmaktadır. İlişkiler kurallar ile ifade edilmektedir. Bu aşamada kural çıkartmak için CBA algoritmasında varsayılan (default) olarak tanımlanmış Apriori algoritması kullanılmaktadır [49]. Algoritma X⇒Y şeklinde kurallar oluşturmaktadır. Kuralın sol tarafı olan X, veri setinde birlikte bulunan örnekleri göstermektedir. Sağ tarafı olan Y ise sadece sınıf değişkenini (etiketi) ifade etmektedir [24]. Kurallar oluşturulurken minimum destek ve minimum güven parametreleri kullanılır. X⇒Y kuralı için; destek değeri X ve Y'nin aynı anda birlikte sağlanması, güven değeri ise X'in sağlanması durumunda Y'nin sağlanması ihtimalidir. Birliktelik kuralları algoritmalarında farklı olarak CBA algoritmasında kurallar oluşturulurken minimum destek kuralın tamamı için değil yalnızca kuralın sol tarafı için kontrol edilmektedir. Kurallar güven, destek ve boyuta göre sıralanır. Daha sonra en iyi kuralları seçmek için zayıf kurallar budanır. Budama işlemi için CBA algoritması kapsamında M1 ve M2 algoritmaları geliştirilmiştir [23]. CBA algoritmasında varsayılan olarak M1 kullanılmaktadır. Budama işlemi bittikten sonra CBA-RG aşaması sonlanmış olur.

CBA-CB, CBA algoritmasının özüdür. Çünkü CBA algoritmasından beklenen ilişkili kuralların etiketli olmasıdır. CBA-RG aşamasında budanıp gereksiz kurallar kaldırıldıktan sonra bu aşamada kuralların etiketlenmesi gerçekleşmektedir. En iyi sınıflandırıcıyı üretmek için eğitim verisindeki tüm kurallar değerlendirilmektedir. En az sayıda hata veren kural, sınıflandırıcı olarak seçilmektedir. Eğitim veri seti ile model eğitimi gerçekleşmektedir. Ardından model test veri seti ile verilen herhangi bir kuralın sınıfını tahmin etmektedir. Bu kural ile eşleşen ilk kuralın sınıfı, tahmin edilen sınıf olarak atanmaktadır [23].

Şekil 2.2.'nin ikinci adımında modellerin birleştirilmesi gerçekleşmektedir. Her bir uçtaki cihazda model eğitimi gerçekleştirildikten sonra modeller sunucuya gönderilmektedir. Eğitilmiş bu modellerin sunucuda birleştirilmesi için du-CBA içerisinde bir modül geliştirilmiştir. Modül ile tüm uçlarda oluşturulan modeller ve bu modellerin eğitiminde

kullanılan veri içerisindeki örnek sayı bilgisi (N) alınmaktadır. Uçlarda eğitilen modeller CBA algoritması ile eğitildiği için etiketli kurallardan oluşmaktadır. Aslında modellerin birleştirilmesi modeller ile birlikte gelen kuralların birleştirilmesi anlamına gelmektedir. Modeller birleştirilirken gelen tüm kurallar kontrol edilmekte ardından birleştirme işlemi yapılmaktadır.

Modellerin birleştirilmesi işlemi, kuralların destek ve güven değerlerinin güncellenmesi ve tekrar sıralanması ile gerçekleşmektedir. Farklı modellerden gelen aynı etikete sahip aynı kurallar için tüm veri içerisinde bulunma sıklığı değişeceğinden dolayı kurala ait destek ve güven değeri güncellenmektedir. Farklı etiketlenmiş aynı kurallar için ise önce bulunma sıklıkları kontrol edilmektedir. Sonra destek değeri büyük olan kural elde tutulmaktadır. Güncelleme işlemi yapıldıktan sonra yeni güven ve destek değerlerine sahip olan her kural önce güven değerine göre sıralanmaktadır. Güven değeri aynı ise destek değerine göre ikincil sıralama yapılmaktadır. Destek değerlerinin de aynı olması durumunda, önce gelen kural daha önde yer almaktadır. Elde edilen yeni kural listesi sunucuda bulunan nihai modeli oluşturmaktadır.

Yukardaki anlatılanlara göre modellerin birleştirilesi için destek ve güven değerlerine göre işlem yapılmaktadır. Geliştirilen algoritma kapsamında destek ve güven değerlerinin güncellenmesi için formüller geliştirilmiştir.

CBA algoritması ile model eğitildikten sonra oluşan kuralların zayıf ya da güçlü kurallar olup olmadığını göstermek için destek ve güven değerlerine bakılmaktadır. Destek (Support): Bir ilişkinin veri seti içinde tekrarlanma oranıdır. Güven (Confidence) ise X⇒Y kuralı için, X'in bulunduğu ilişkide Y'nin bulunma olasılığıdır.
Kural oluşturulurken destek değerini belirlemede kullanılan formül (2.1) nolu denklemde, güven değerini belirlemede kullanılan formül (2.2) nolu denklemde gösterilmiştir.

$$Destek\ (X \Rightarrow Y) = \frac{\text{Frekans (X,Y)}}{N} \qquad (2.1)\ [50]$$

$$Güven\ (X \Rightarrow Y) = \frac{\text{Frekans (X,Y)}}{\text{Frekans (X)}} \qquad (2.2)\ [50]$$

X kuralın sol tarafını (lhs: Left hand side), Y ise kuralın sağ tarafını (rhs: Right hand side), yani kuralın etiketini göstermektedir. İstemcilerin ve sunucunun birlikte çalıştığı sistemler için, uçta çalışan cihaz sayısı bilinmediğinden uç sayısı n olarak ifade edilmiştir. n tane uçta bulunan cihazlar için eğitilen her bir modeldeki kurallarının destek değeri yukarıda gösterilen (2.1) nolu denkleme göre hesaplanır. Güven değeri ise (2.2) nolu denkleme göre hesaplanır. Her uçtan gelecek modellerin sunucuda birleştirilme işlemi kuralların destek ve güven değerlerinin güncellenmesi ve tekrar sıralanması ile gerçekleşmektedir. Model birleştirme işlemini gerçekleştirmek amacıyla verinin destek değerini güncelleyen matematiksel formül (2.5) nolu denklemde, güven değerlerini güncelleyen formül ise (2.9) nolu denklemde yer almaktadır. Fomüllerde bilinmeyen değerler yalnız bırakılarak, bilinen değerler ile yeni destek ve güven değerleri bulunmaya çalışılmıştır. N model eğitiminde kullanılan veri seti içinde bulunan toplam örnek sayısıdır.

$$Destek1 = \frac{\text{Frekans(X1,Y1)}}{N1} \quad Destek2 = \frac{\text{Frekans(X2,Y2)}}{N2} \quad \dots \quad Destekn = \frac{\text{Frekans(Xn,Yn)}}{Nn} \tag{2.3}$$

$$Destek_{birleştirme} = \frac{\text{Frekans(X1,Y1)+ Frekans(X2,Y2)+ Frekans(X3,Y3)+}\cdots\text{+ Frekans(Xn,Yn)}}{N1+N2+N3+\cdots+Nn} \tag{2.4}$$

$$Destek_{birleştirme} = \frac{\text{(Destek1}*\text{N1)+(Destek2}*\text{N2)+(Destek3}*\text{N3)+}\cdots\text{+(Destekn}*\text{Nn)}}{N1+N2+N3+\cdots+Nn} \tag{2.5}$$

(2.3) nolu denklemde, 1'den n'ye kadar uçta bulunan her cihazdan gelen kuralların destek formülleri yer almaktadır. Formüldeki N ve destek değeri bilinmektedir. Model birleştirilirken kuralların destek değerleride (2.4) nolu denklemde gösterildiği gibi birleştirilmektedir. (2.4) nolu denklemde 1'den n'ye tüm Frekans (X,Y) değerleri yerine, (2.3) nolu denkleme bakılarak elde edilen Destek*N değeri yazılmaktadır. Tüm bu işlemlerin sonucunda verinin destek değerini güncelleyen (2.5) nolu denklemde yer alan formül elde edilmektedir.

$$Güven = \frac{\text{Frekans(X,Y)}}{Frekans(X)} => \frac{\text{Destek}*\text{N}}{Güven} = Frekans(X) \tag{2.6}$$

$$Güven1 = \frac{\text{Frekans(X1,Y1)}}{Frekans(X1)} \quad Güven2 = \frac{\text{Frekans(X2,Y2)}}{Frekans(X2)} \quad \dots \quad Güvenn = \frac{\text{Frekans(Xn,Yn)}}{Frekans(Xn)} \tag{2.7}$$

$$Güven_{birleştirme} = \frac{\text{Frekans(X1,Y1)+ Frekans(X2,Y2)+ Frekans(X3,Y3)+⋯+ Frekans(Xn,Yn)}}{\text{Frekans(X1)+ Frekans(X2)+ Frekans(X3)+⋯+ Frekans(Xn)}} \quad (2.8)$$

$$Güven_{birleştirme} = \frac{(\text{Destek1}*\text{N1})+(\text{Destek2}*\text{N2})+(\text{Destek3}*\text{N3})+⋯+(\text{Destken}*\text{Nn})}{\frac{\text{Destek1}*\text{N1}}{Güven1}+\frac{\text{Destek2}*\text{N2}}{Güven2}+\frac{\text{Destek3}*\text{N3}}{Güven3}+⋯+\frac{\text{Destekn}*\text{Nn}}{Güvenn}} \quad (2.9)$$

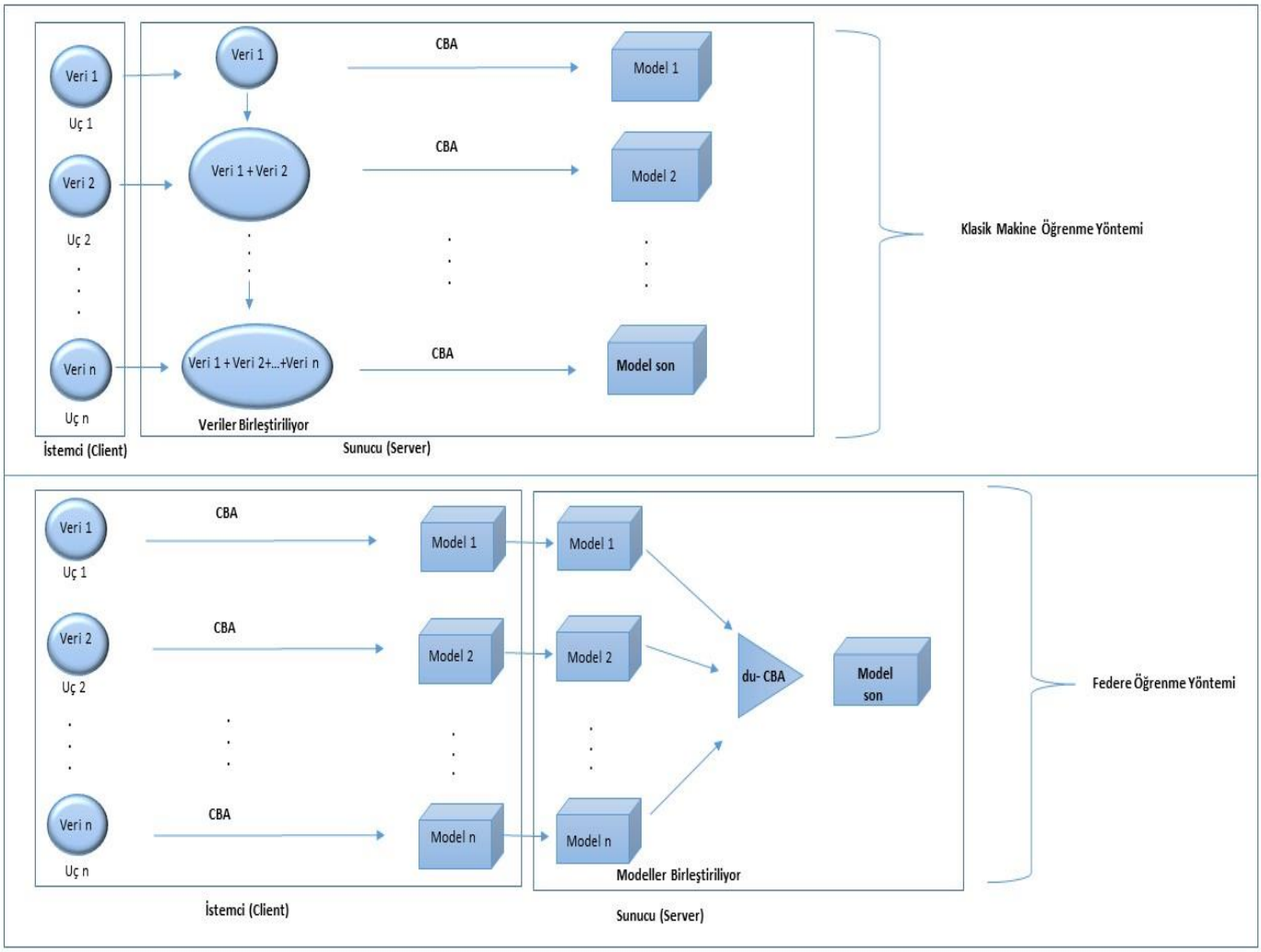
(2.3) nolu denklemde bulunan destek formülündeki değerler, (2.6) nolu denklemde bulunan güven formülünde yerine yazarak Frekans (X) değeri elde edilmektedir. (2.7) nolu denklemde, 1'den n'ye kadar uçta bulunan her cihazdan gelen kuralların güven formülleri yer almaktadır. Model birleştirilirken kuralların güven değerleride (2.8) nolu denklemde gösterildiği gibi birleştirilmektedir. (2.9) nolu denklem için 1'den n'ye formülün pay kısmı olan tüm Frekans (X,Y) değerleri yerine, (2.3) nolu denkleme bakılarak elde edilen Destek*N değeri yazılmıştır. Payda kısmındaki Frekans (X) değerleri ise (2.6) nolu denkleme bakılarak düzenlenmiştir. Tüm bu işlemlerin sonucunda verinin destek değerini güncelleyen (2.9) nolu denklemde yer alan formül elde edilmektedir.

3. BULGULAR ve TARTIŞMA

Bu bölümde federe öğrenme mimarisi ile oluşturulan prototip uygulamadan, deneylerin gerçekleştirildiği ortamdan, deneylerin değerlendirilme ölçütlerinden ve elde edilen sonuçlardan bahsedilmektedir. Deneylerde, CBA ve bu makalede önerilen du-CBA algoritmaları ile eğitilen modellerin eğitim süreleri ve performansları kıyaslanmıştır.

3.1. Prototip Uygulama

Tez kapsamında geliştirilen algoritmanın gerçekleştirildiği ve kıyaslandığı simülasyon ortamı Şekil 3.1'de görselleştirilmiştir. du-CBA algoritması federe öğrenme mimarisi ile öğrenme gerçekleştirmektedir ve ilişkisel sınıflandırma algoritmasıdır.



Şekil 3.1. Benzetim ortamında kıyaslanan klasik makine öğrenmesi yöntemi ve federe öğrenme yönteminin şekil ile gösterimi.

Şekil 3.1'de simülasyon ortamında iki farklı mimari yöntem ile gerçekleştirilen model eğitimi gösterilmiştir. İlk model eğitiminde klasik öğrenme yöntemi kullanılmıştır yani istemcilerden sunucuya veriler aktarılmış ve bu verilerden model eğitimi gerçekleşmiştir. Bu yöntemde sunucuya her uçtan veri gelmekte yani veriler sunucuda toplanmaktadır. Toplanan veriler ile eğitim yapılarak model oluşmaktadır. İkinci yöntem ise federe öğrenme mimarisidir. Bu mimari ile istemcilerde model eğitilip sunucuda modellerin birleştirilmesi ile nihai modeli elde edilmektedir. Simülasyon ortamında klasik öğrenme ile model eğitimini gerçekleştirmek için CBA algoritması, federe öğrenme mimarisini gerçekleştirmek için ise geliştirilen du-CBA algoritması kullanılmıştır. Bunun için öncelikle kullanılan veri setleri eğitim ve test setine ayrılmıştır. Eğitim setinde bulunan veriler sanki farklı uçlardan geliyormuş gibi varsayılmış ve uç sayısı kadar parçaya neredeyse eşit sayıda veri içerecek şekilde rastgele ayrılmıştır. Örneğin uç sayısı 2 kabul edildiğinde, eğitim seti neredeyse eşit iki parçaya rastgele ayrılmıştır. Buna göre; CBA algoritması test edilirken bu 2 parça birleştirilerek model eğitilmiş ve ardından test gerçekleştirilmiştir. Önerilen du-CBA algoritmasında ise bu 2 parçadan ayrı ayrı 2 model eğitilmiştir. Veriden bağımsız olarak uçlardan sunucuya sadece modellerin gönderildiği simüle edilmiştir. Modeller etiketli kurallardan oluşmaktadır. Her bir uçtan sunucuya tek başına anlamsız verilerden oluşan etiketli kurallar yani modeller gönderilmiş ve bu modeller birleştirilmiştir. Aslında modellerin birleştirilmesi modeller ile birlikte gelen kuralların birleştirilmesi anlamına gelmektedir. Sunucuda birleştirme işlemi ardından yeni bir model elde edilmiştir. Bu model nihai modeldir. Elde edilen nihai model ile testler gerçekleştirilmiştir.

Şekil 3.1'de uçta çalışan cihaz sayısı bilinmediği için uç sayısı n olarak ifade edilmiştir. Farklı uç sayıları ile çalışmayı gözlemlemek için bahsedilen işlemler uç sayısı 4, 8, 16, 32, 64 ve 128 olarak varsayıldığı durumlar için de tekrarlanmıştır. Test sonuçları paylaşılmış ve kıyaslamalar gerçekleştirilmiştir.

3.2. Deney Ortamı

Deneyler için kullanılan bilgisayar, 11.Nesil i5-1135G7 işlemci ve 16 GB DDR3 belleğe sahiptir. Federe öğrenme mimarisi kapsamında geliştirilen algoritma du-CBA, CBA algoritması üzerine inşa edilmiştir ve PyCharm IDE [51] ortamında geliştirilmiştir. CBA ve du-CBA algoritmaları uygulanırken pyArc [52] modülü kullanılmış, ilişkilerin çıkarılması için ise Apriori [49] algoritması tercih edilmiştir. İlişkisel sınıflandırma yöntemi ile oluşturulan kural sayısını ve tahmin doğruluğunu kontrol etmek için kullanılan değerler; minimum destek ve minimum güvendir [34]. Hem önerilen algoritma (du-CBA) hem de CBA için gerekli destek ve

güven değerleri sırasıyla 0.2 ve 0.5 olarak belirlenmiştir. Bu değerler literatürde varsayılan değerlerdir [53-55]. Tüm bu işlemlerden sonra deneysel ortam hazır hale getirilmiştir.

3.3. Performans Ölçütleri

Geliştirilen algoritma du-CBA ve klasik CBA algoritmasının sınıflandırma doğruluğunu, verimliliğini değerlendirmek için farklı başarı karşılaştırma ölçütlerinden faydalanılmıştır. Performans değerlendirmeleri için karışıklık-hata matrisi kullanılmaktadır. Tablo 3.1'de karışıklık matrisi tablosu yer almaktadır. Bu matriste satırlar gerçek sınıf değerlerini, sütunlar ise tahmin edilen değerleri göstermektedir. Matriste Doğru Pozitif, DP (True Positive, TP), Doğru Negatif, DN (True Negative, TN), Yanlış Pozitif, YP (False Positive, FP) ve Yanlış Negatif, YN (False Negative, FN) ifadeleri yer almaktadır. TP ve TN doğru şekilde sınıflandırılan pozitif ve negatif değerlerin sayısını, FP ve FN ise yanlış şekilde sınıflandırılan pozitif ve negatif değerlerin sayısını göstermektedir [56, 57].

Tablo 3.1. Karmaşıklık Matrisi.

	Pozitif Tahmin (+)	**Negatif Tahmin (-)**
Gerçek Pozitif (+)	DP	YN
Gerçek Negatif (-)	YP	DN

Bu çalışmada da doğruluk (accuracy-acc), kesinlik (precision-p), duyarlılık (recall-r), F1 ölçütü (F measure-f) olmak üzere 4 farklı karşılaştırma ölçütü performans kıyaslaması yapabilmek için belirlenmiştir. du-CBA ve CBA ile eğitilen bütün modellerin ortalama skorları, standart sapma değerleri raporlanarak modellerin performanslarının değerlendirilmesinde birer ölçüt olarak kullanılmıştır.

Doğruluk: Modelde doğru tahmin edilen tüm alanların, tüm veri setine oranı ile bulunmaktadır. (3.1) nolu denklem ile hesaplanır.

$$Doğruluk = \frac{DP+DN}{DP+DN+YP+YN} \tag{3.1}$$

Kesinlik: Doğru pozitif tahminlerin sayısının bütün pozitif değerlere oranını ifade etmektedir.

(3.2) nolu denklem ile hesaplanır.

$$Kesinlik = \frac{DP}{DP+YP} \quad (3.2)$$

Duyarlılık: Gerçek değeri pozitif olup pozitif olarak sınıflandırılan değerlerin veri setindeki gerçek pozitif değerlerin sayısına bölünmesi ile elde edilmektedir. (3.3) nolu denklem ile hesaplanır.

$$Duyarlılık = \frac{DP}{DP+YN} \quad (3.3)$$

F Skor: Kesinlik ve duyarlılık değerlerinin harmonik ortalamasıdır. (3.4) nolu denklem ile hesaplanır.

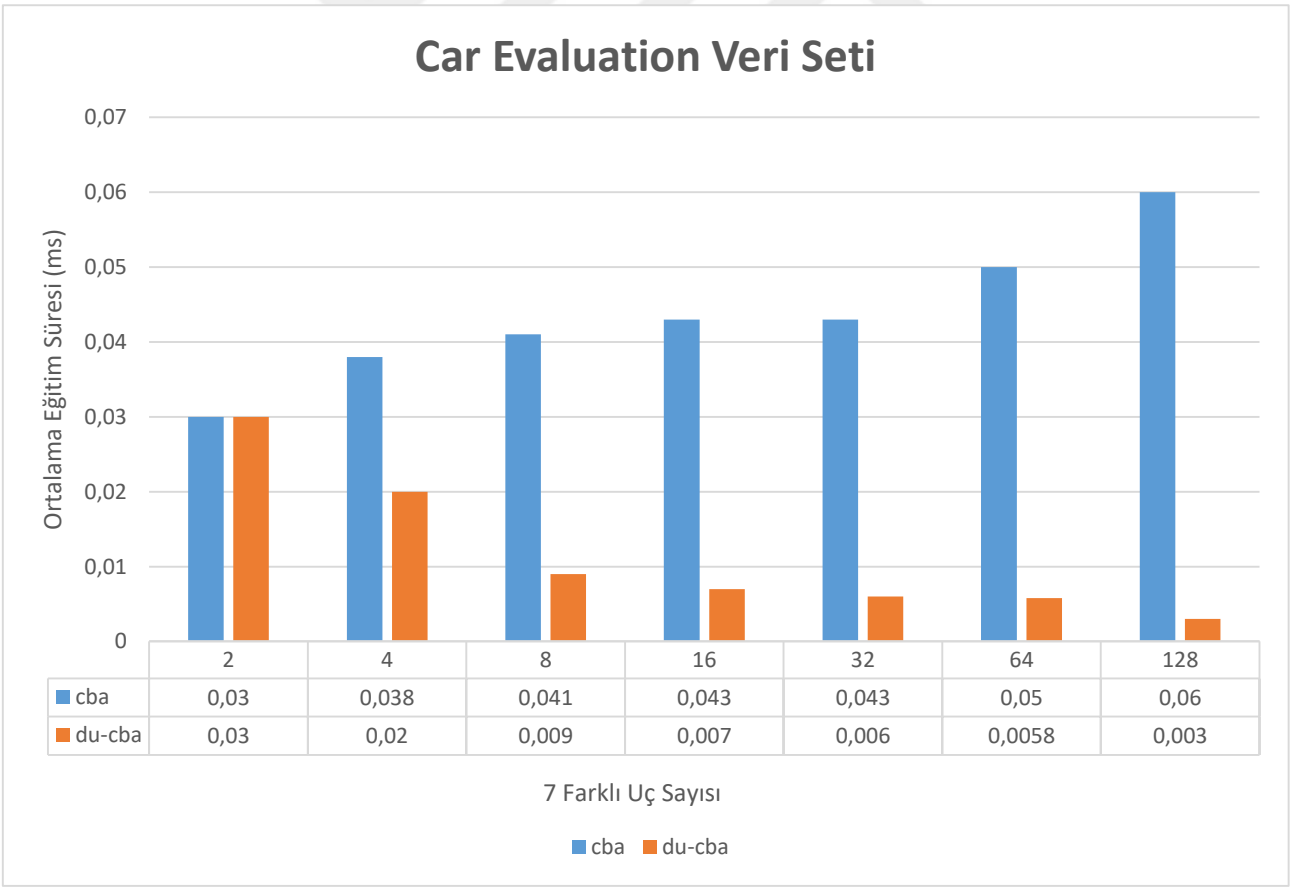
$$\text{F Score} = 2 \text{ x } \frac{Kesinlik\ x\ Duyarlılık}{Kesinlik + Duyarlılık} \quad (3.4)$$

3.4. Deney Sonuçları

Simülasyon ortamında yapılan testler 5 kez tekrar edilerek çalıştırılmıştır. Geliştirilen algoritma içerisinde veri seti rastgele bölündüğü için her test sonucunda farklı değerler ortaya çıkmıştır. Testler sonucunda elde edilen değerlerin ortalaması alınmıştır. Her bir veri seti ve varsayılan her bir uç sayısı için doğruluk (accuracy), kesinlik (precision), duyarlılık (recall), F1 ölçütü değerlerinin ortalama sonucu ve bu değerlere göre hesaplanan strandart sapma sonucu tablo üzerinde gösterilmiştir. Modellerin eğitim sürelerinin ortalama değerleri ise şekil üzerinde gösterilmiştir.

Uç sayısı 2, 4, 8, 16, 32, 64 ve 128 olarak varsayılan yedi durum için, beş farklı veri seti ile CBA ve du-CBA algoritmalarının eğitim sürelerinin kıyaslamasını gösteren grafikler aşağıda yer almaktadır.

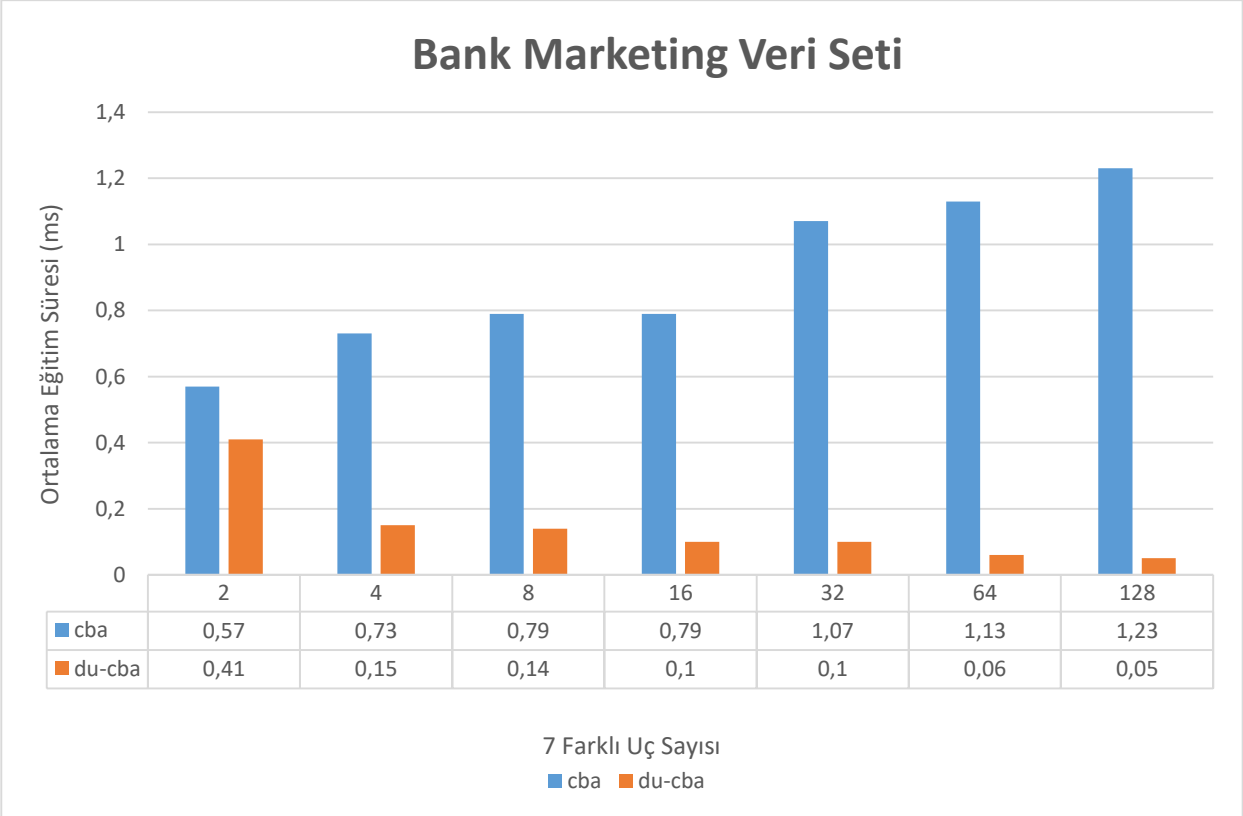
Şekil 3.2'de 7 farklı uç sayısı kullanılarak car evaluation veri seti için ortalama eğitim süresi gösterilmektedir. Şekillerde uç sayısı yatay eksende, eğitim zamanı dikey eksende gösterilmektedir. Şekil üzerinde her bir uç sayısına özel çubuklardan ilki klasik yöntem ile modelin eğitim süresini ifade etmektedir. CBA ile model eğitimi için önce birinci uçtan gelen veri ile model eğitilmiştir. Daha sonra ikinci uçtan gelen veri birinci uçtan gelen veri ile birleştirilip tekrar model eğitilmiştir. Her uçtan gelen verinin, mevcut veriler ile birleştirilerek model eğitilmesi işlemi, son uç sayısına kadar devam etmektedir. Grafikte modelin eğitimi için harcanan zaman gösterilmektedir. Şekil üzerinde her bir uç sayısına özel çubuklardan ikinci çubuk ise federe öğrenme yöntemi ile modelin eğitim süresini ifade etmektedir. du-CBA algoritması kullanılarak ilk uçtan son uca kadar modeller eğitilmiş ve eğitilen tüm modeller birleştirilmiştir. Grafik, modellerin eğitimi ve birleştirilme işlemi için harcanan zamanı göstermektedir. Grafiğin altında tablo şeklinde iki algoritmanın her uç için ayrı ayrı eğitim süresi değerleri paylaşılmıştır.



	2	4	8	16	32	64	128
cba	0,03	0,038	0,041	0,043	0,043	0,05	0,06
du-cba	0,03	0,02	0,009	0,007	0,006	0,0058	0,003

Şekil 3.2. CBA ve du-CBA algoritmalarının Car Evaluation veri setini kullanarak 2, 4, 8, 16, 32, 64 ve 128 olarak belirlenen yedi farklı uç sayıları için tek tek modellerin eğitim sürelerinin karşılaştırılması.

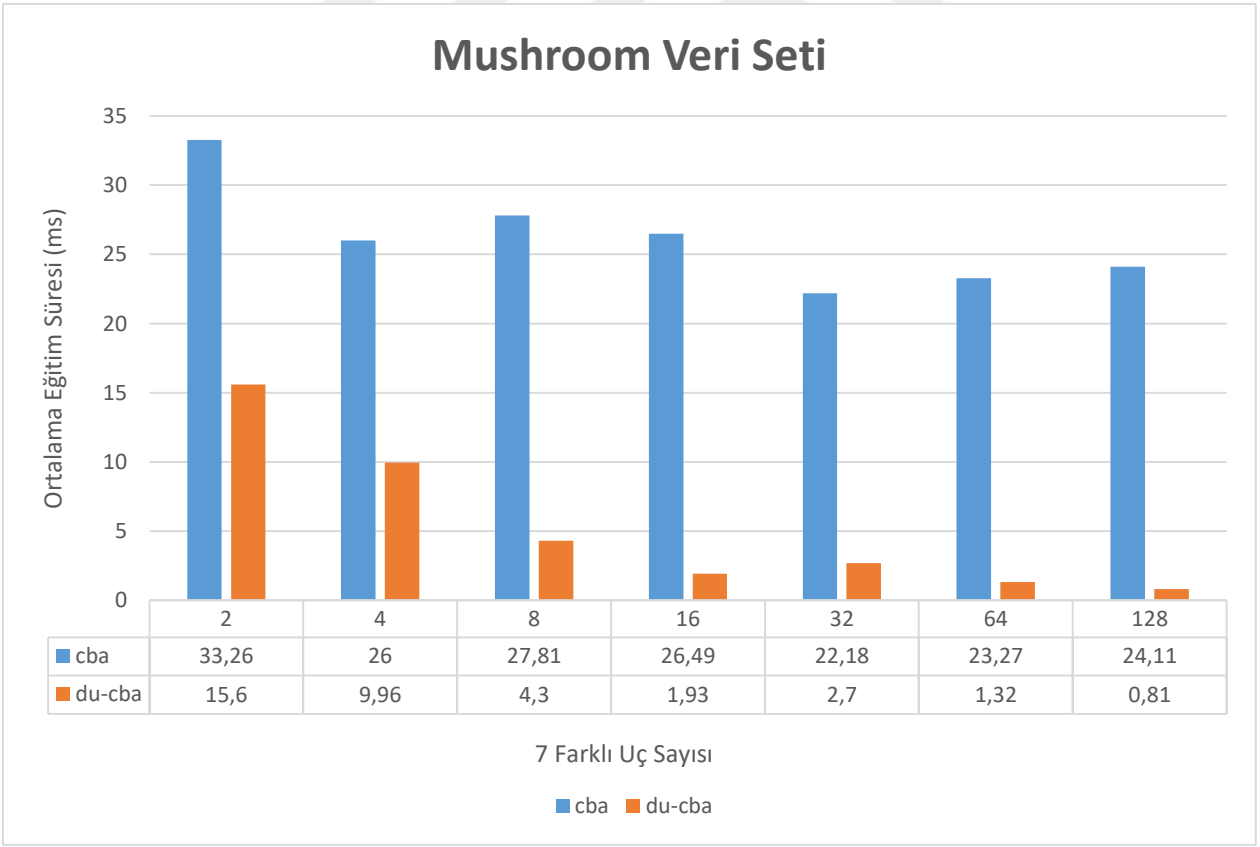
Şekil 3.3'te 7 farklı uç sayısı kullanılarak bank marketting veri seti için ortalama eğitim süresi gösterilmektedir. Şekillerde uç sayısı yatay eksende, eğitim zamanı dikey eksende gösterilmektedir. Şekil üzerinde her bir uç sayısına özel çubuklardan ilki klasik yöntem ile modelin eğitim süresini ifade etmektedir. CBA ile model eğitimi için önce birinci uçtan gelen veri ile model eğitilmiştir. Daha sonra ikinci uçtan gelen veri birinci uçtan gelen veri ile birleştirilip tekrar model eğitilmiştir. Her uçtan gelen verinin, mevcut veriler ile birleştirilerek model eğitilmesi işlemi son uç sayısına kadar devam etmektedir. Grafikte modelin eğitimi için harcanan zaman gösterilmektedir. Şekil üzerinde her bir uç sayısına özel çubuklardan ikşnci çubuk ise federe öğrenme yöntemi ile modelin eğitim süresini ifade etmektedir. du-CBA algoritması kullanılarak ilk uçtan son uca kadar modeller eğitilmiş ve eğitilen tüm modeller birleştirilmiştir. Grafik, modellerin eğitimi ve birleştirilme işlemi için harcanan zamanı göstermektedir. Grafiğin altında tablo şeklinde iki algoritmanın her uç için ayrı ayrı eğitim süresi değerleri paylaşılmıştır.



	2	4	8	16	32	64	128
cba	0,57	0,73	0,79	0,79	1,07	1,13	1,23
du-cba	0,41	0,15	0,14	0,1	0,1	0,06	0,05

Şekil 3.3. CBA ve du-CBA algoritmalarının Bank Marketing veri setini kullanarak 2, 4, 8, 16, 32, 64 ve 128 olarak belirlenen yedi farklı uç sayıları için tek tek modellerin eğitim sürelerinin karşılaştırılması.

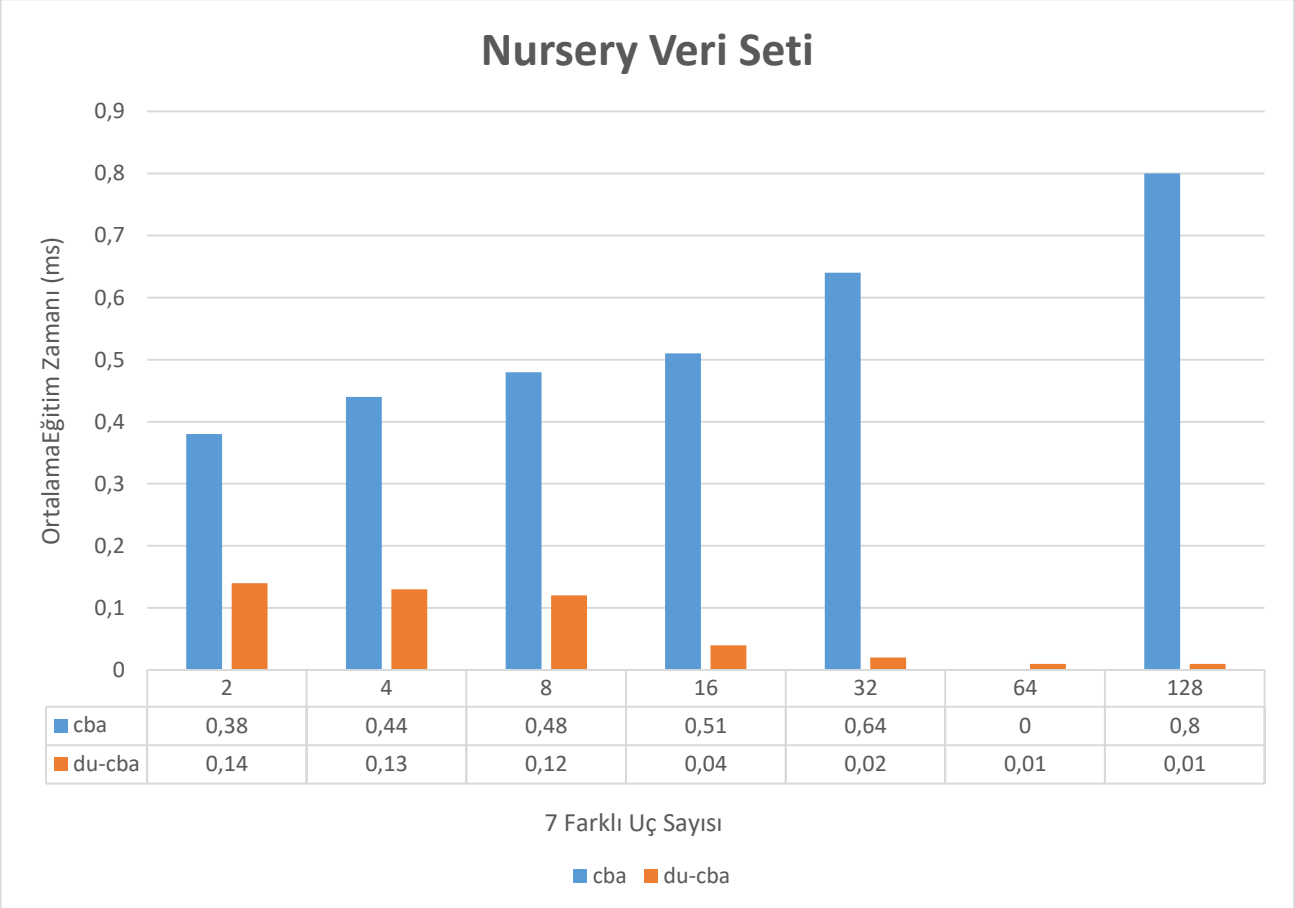
Şekil 3.4'te 7 farklı uç sayısı kullanılarak mushroom veri seti için ortalama eğitim süresi gösterilmektedir. Şekillerde uç sayısı yatay eksende, eğitim zamanı dikey eksende gösterilmektedir. Şekil üzerinde her bir uç sayısına özel çubuklardan ilki klasik yöntem ile modelin eğitim süresini ifade etmektedir. CBA ile model eğitimi için önce birinci uçtan gelen veri ile model eğitilmiştir. Daha sonra ikinci uçtan gelen veri birinci uçtan gelen veri ile birleştirilip tekrar model eğitilmiştir. Her uçtan gelen verinin, mevcut veriler ile birleştirilerek model eğitilmesi işlemi son uç sayısına kadar devam etmektedir. Grafikte modelin eğitimi için harcanan zaman gösterilmektedir. Şekil üzerinde her bir uç sayısına özel çubuklardan ikinci çubuk ise federe öğrenme yöntemi ile modelin eğitim süresini ifade etmektedir. du-CBA algoritması kullanılarak ilk uçtan son uca kadar modeller eğitilmiş ve eğitilen tüm modeller birleştirilmiştir. Grafik, modellerin eğitimi ve birleştirilme işlemi için harcanan zamanı göstermektedir. Grafiğin altında tablo şeklinde iki algoritmanın her uç için ayrı ayrı eğitim süresi değerleri paylaşılmıştır.



	2	4	8	16	32	64	128
cba	33,26	26	27,81	26,49	22,18	23,27	24,11
du-cba	15,6	9,96	4,3	1,93	2,7	1,32	0,81

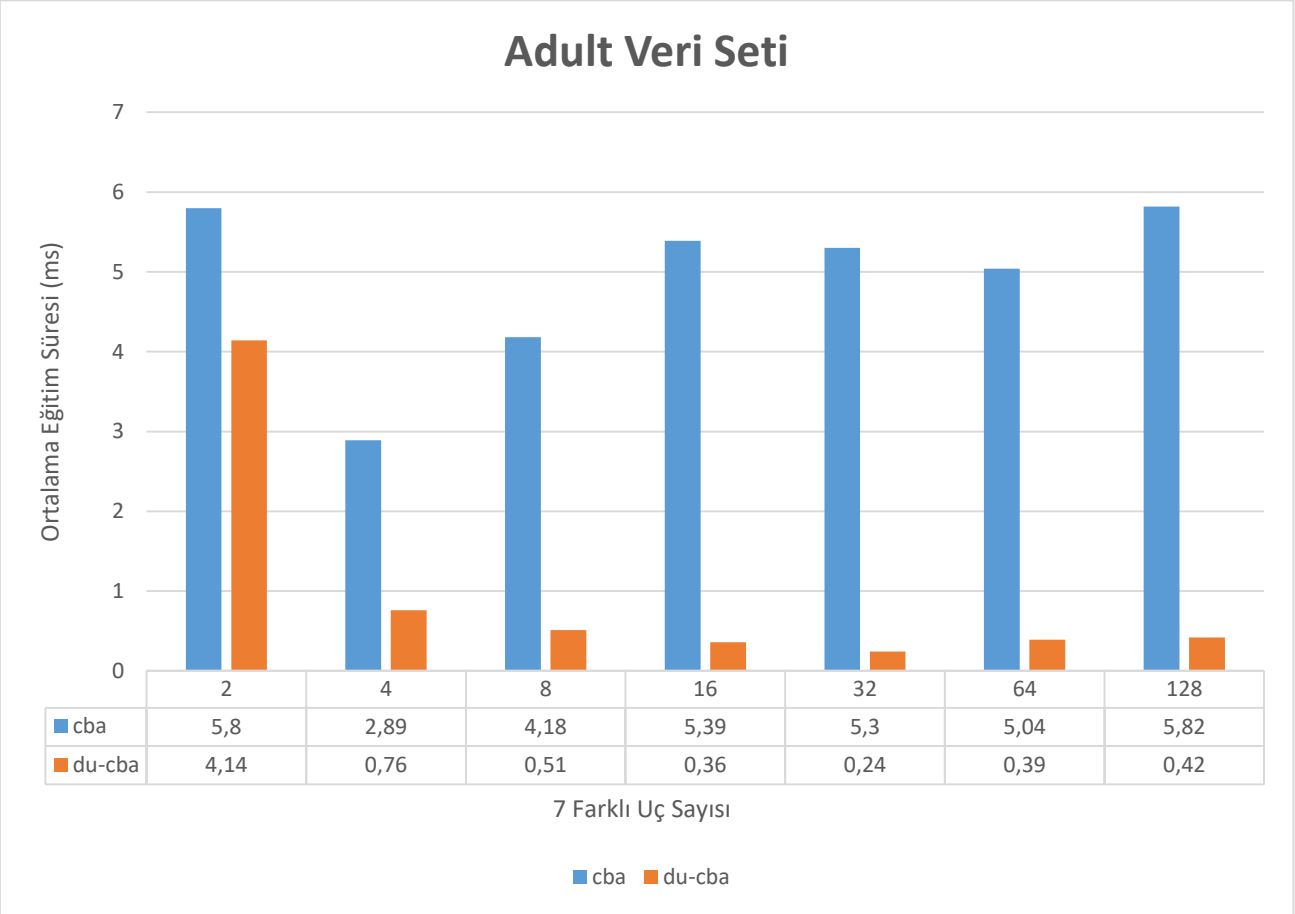
Şekil 3.4. CBA ve du-CBA algoritmalarının Mushroom veri setini kullanarak 2, 4, 8, 16, 32, 64 ve 128 olarak belirlenen yedi farklı uç sayıları için tek tek modellerin eğitim sürelerinin karşılaştırılması.

Şekil 3.5'te 7 farklı uç sayısı kullanılarak nursery veri seti için ortalama eğitim süresi gösterilmektedir. Şekillerde uç sayısı yatay eksende, eğitim zamanı dikey eksende gösterilmektedir. Şekil üzerinde her bir uç sayısına özel çubuklardan ilki klasik yöntem ile modelin eğitim süresini ifade etmektedir. CBA ile model eğitimi için önce birinci uçtan gelen veri ile model eğitilmiştir. Daha sonra ikinci uçtan gelen veri birinci uçtan gelen veri ile birleştirilip tekrar model eğitilmiştir. Her uçtan gelen verinin, mevcut veriler ile birleştirilerek model eğitilmesi işlemi son uç sayısına kadar devam etmektedir. Grafikte modelin eğitimi için harcanan zaman gösterilmektedir. Şekil üzerinde her bir uç sayısına özel çubuklardan ikinci çubuk ise federe öğrenme yöntemi ile modelin eğitim süresini ifade etmektedir. du-CBA algoritması kullanılarak ilk uçtan son uca kadar modeller eğitilmiş ve eğitilen tüm modeller birleştirilmiştir. Grafik, modellerin eğitimi ve birleştirilme işlemi için harcanan zamanı göstermektedir. Grafiğin altında tablo şeklinde iki algoritmanın her uç için ayrı ayrı eğitim süresi değerleri paylaşılmıştır.



Şekil 3.5. CBA ve du-CBA algoritmalarının Nursery veri setini kullanarak 2, 4, 8, 16, 32, 64 ve 128 olarak belirlenen yedi farklı uç sayıları için tek tek modellerin eğitim sürelerinin karşılaştırılması.

Şekil 3.6'da 7 farklı uç sayısı kullanılarak adult veri seti için ortalama eğitim süresi gösterilmektedir. Şekillerde uç sayısı yatay eksende, eğitim zamanı dikey eksende gösterilmektedir. Şekil üzerinde her bir uç sayısına özel çubuklardan ilki klasik yöntem ile modelin eğitim süresini ifade etmektedir. CBA ile model eğitimi için önce birinci uçtan gelen veri ile model eğitilmiştir. Daha sonra ikinci uçtan gelen veri birinci uçtan gelen veri ile birleştirilip tekrar model eğitilmiştir. Her uçtan gelen verinin, mevcut veriler ile birleştirilerek model eğitilmesi işlemi son uç sayısına kadar devam etmektedir. Grafikte modelin eğitimi için harcanan zaman gösterilmektedir. Şekil üzerinde her bir uç sayısına özel çubuklardan ikinci çubuk ise federe öğrenme yöntemi ile modelin eğitim süresini ifade etmektedir. du-CBA algoritması kullanılarak ilk uçtan son uca kadar modeller eğitilmiş ve eğitilen tüm modeller birleştirilmiştir. Grafik, modellerin eğitimi ve birleştirilme işlemi için harcanan zamanı göstermektedir. Grafiğin altında tablo şeklinde iki algoritmanın her uç için ayrı ayrı eğitim süresi değerleri paylaşılmıştır.



	2	4	8	16	32	64	128
cba	5,8	2,89	4,18	5,39	5,3	5,04	5,82
du-cba	4,14	0,76	0,51	0,36	0,24	0,39	0,42

Şekil 3.6. CBA ve du-CBA algoritmalarının Adult veri setini kullanarak 2, 4, 8, 16, 32, 64 ve 128 olarak belirlenen yedi farklı uç sayıları için tek tek modellerin eğitim sürelerinin karşılaştırılması.

Grafik sonuçlarına göre tüm durumlar için federe öğrenme yönteminin klasik öğrenme yönteminden daha hızlı çalışma zamanına sahip olduğu sonucuna varılmıştır.

CBA ve du-CBA algoritmaları kullanılarak eğitilen bütün modellerin performans ölçütlerinin ortalama değerleri ve standart sapma değerleri tablolar halinde paylaşılmıştır. Bu değerler algoritmaların performanslarının değerlendirilmesinde kullanılmıştır.

Tablo 3.2'de CBA ve du-CBA algoritmalarının beş farklı veri seti üzerindeki başarı ölçütleri gösterilmiştir. Test edilen tüm durumlar için algoritmaların ortalama doğruluk (average accuracy), ortalama kesinlik (average precision), ortalama duyarlılık (average recall) ve ortalama F1 ölçütü (average F1 skor) değerleri karşılaştırılmış ve sonuçları paylaşılmıştır. Elde edilen sonuçlara göre iki yöntemin birbirlerine göre başarı ölçütleri bakımından belirgin bir üstünlüğü gözlenmemiştir.

Tablo 3.3'te CBA ve du-CBA algoritmaları ile eğitilen her bir modelin, eğitilmesinden elde edilen performans ölçütlerine ait değerlerin, standart sapmaları yer almaktadır. İstatistikte standart sapma değerinin küçük olması verilerin ortalama değere daha yakın şekilde dağıldığı anlamına gelmektedir. Tablo 3.3'e göre standart sapma değerlerinin düşük olması CBA ve du-CBA ile elde edilen her bir karşılaştırma ölçütü değerinin ortalama değere daha yakın şekilde dağılım gösterdiği anlamına gelmektedir. Bu istenilen bir durumdur.

Tablo 3.2. Beş farklı veri kümesi için CBA ve du-CBA algoritmaları ile eğitilen modellerin performans ölçülerinin ortalama değerleri.

Uç Sayısı	Veri Seti	CBA				du-CBA			
		Ortalama Doğruluk (Average Accuracy)	Ortalama Kesinlik (Average Precision)	Ortalama Duyarlılık (Average Recall)	Ortalama F1	Ortalama Doğruluk (Average Accuracy)	Ortalama Kesinlik (Average Precision)	Ortalama Duyarlılık (Average Recall)	Ortalama F1
2	Car Evaluation	0,8	0,78	0,8	0,78	0,8	0,76	0,81	0,79
	Bank Marketing	0,88	0,8	0,86	0,84	0,88	0,78	0,88	0,83
	Mushroom	0,98	0,99	0,99	0,99	0,98	0,99	0,99	0,99
	Nursery	0,66	0,52	0,67	0,56	0,66	0,52	0,67	0,56
	Adult	0,77	0,75	0,78	0,74	0,77	0,76	0,77	0,73
4	Car Evaluation	0,79	0,76	0,8	0,77	0,79	0,73	0,79	0,76
	Bank Marketing	0,88	0,78	0,88	0,83	0,88	0,78	0,88	0,83
	Mushroom	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,98
	Nursery	0,65	0,6	0,66	0,6	**0,67**	0,63	0,67	0,64
	Adult	0,77	0,76	0,78	0,74	0,77	0,79	0,78	0,71
8	Car Evaluation	0,79	0,78	0,8	0,78	0,79	0,72	0,79	0,74
	Bank Marketing	0,87	0,77	0,88	0,8	0,87	0,82	0,88	0,8
	Mushroom	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99
	Nursery	0,66	0,6	0,67	0,6	**0,7**	0,7	0,71	0,7
	Adult	0,77	0,75	0,78	0,75	0,77	0,79	0,77	0,7
16	Car Evaluation	0,79	0,78	0,79	0,78	0,72	0,6	0,72	0,63
	Bank Marketing	0,87	0,77	0,87	0,82	0,87	0,77	0,87	0,82
	Mushroom	0,98	0,99	0,99	0,99	0,98	0,99	0,99	0,99
	Nursery	0,66	0,6	0,66	0,6	**0,71**	0,69	0,71	0,7
	Adult	0,78	0,76	0,78	0,75	0,77	0,79	0,78	0,71
32	Car Evaluation	0,79	0,77	0,8	0,78	0,71	0,6	0,71	0,61
	Bank Marketing	0,87	0,79	0,88	0,83	0,87	0,79	0,88	0,83
	Mushroom	0,98	0,99	0,99	0,99	0,95	0,95	0,95	0,95
	Nursery	0,66	0,6	0,66	0,6	**0,71**	0,71	0,72	0,72
	Adult	0,78	0,76	0,74	0,72	0,76	0,79	0,77	0,68
64	Car Evaluation	0,79	0,77	0,8	0,78	0,7	0,6	0,71	0,61
	Bank Marketing	0,88	0,77	0,88	0,83	0,88	0,77	0,88	0,83
	Mushroom	0,98	0,99	0,99	0,99	0,95	0,95	0,95	0,95
	Nursery	0,66	0,61	0,66	0,61	**0,71**	0,7	0,72	0,71
	Adult	0,78	0,76	0,78	0,75	0,76	0,79	0,77	0,67
128	Car Evaluation	0,8	0,77	0,81	0,79	0,73	0,69	0,73	0,68
	Bank Marketing	0,88	0,78	0,89	0,83	0,88	0,78	0,89	0,83
	Mushroom	0,98	0,99	0,99	0,99	0,91	0,91	0,91	0,91
	Nursery	0,65	0,6	0,65	0,6	**0,75**	0,74	0,75	0,75
	Adult	0,78	0,76	0,78	0,75	0,75	0,78	0,76	0,66

Tablo 3.3. CBA ve du-CBA algoritmalarının beş farklı veri seti ile eğitildiği modellerin performans ölçütlerine ait standart sapma değerleri.

Uç Sayısı	Veri Seti	CBA				du-CBA			
		Acc_Std	P_Std	R_Std	F_Std	Acc_Std	P_Std	R_Std	F_Std
2	Car Evaluation	0,015	0,025	0,018	0,023	0,016	0,035	0,029	0,028
	Bank Marketing	0,008	0,048	0,05	0,025	0,008	0,034	0,008	0,015
	Mushroom	0,004	0	0	0	0,004	0	0	0
	Nursery	0,005	0,029	0,005	0,005	0,005	0,029	0,005	0,005
	Adult	0,005	0,005	0,007	0,008	0,005	0,02	0,004	0,01
4	Car Evaluation	0,016	0,017	0,019	0,015	0,016	0,023	0,02	0,025
	Bank Marketing	0,007	0,017	0,01	0,01	0,007	0,017	0,01	0,01
	Mushroom	0	0,004	0,004	0,004	0	0,004	0,004	0,004
	Nursery	0,008	0	0,007	0	**0,02**	0,04	0,02	0,05
	Adult	0,004	0,005	0,004	0,004	0,005	0,01	0,01	0,01
8	Car Evaluation	0,024	0,033	0,03	0,03	**0,013**	0,023	0,011	0,011
	Bank Marketing	0,011	0,022	0,013	0,05	0,011	0,95	0,013	0,05
	Mushroom	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99
	Nursery	0,01	0	0,01	0	**0,004**	0,008	0,004	0,005
	Adult	0,004	0,004	0,005	0	0,005	0,01	0,004	0,01
16	Car Evaluation	0,79	0,78	0,79	0,78	**0,72**	0,6	0,72	0,63
	Bank Marketing	0,08	0,01	0,008	0,01	0,08	0,01	0,008	0,01
	Mushroom	0,98	0,99	0,99	0,99	0,98	0,99	0,99	0,99
	Nursery	0,01	0	0,01	0	**0**	0,005	0,004	0,005
	Adult	0,008	0,008	0,008	0,008	**0,005**	0,01	0,007	0,01
32	Car Evaluation	0,79	0,77	0,8	0,78	**0,71**	0,6	0,71	0,61
	Bank Marketing	0,008	0,02	0,01	0,01	0,008	0,02	0,01	0,01
	Mushroom	0,98	0,99	0,99	0,99	**0,95**	0,95	0,95	0,95
	Nursery	0	0	0,005	0	0,008	0,01	0,01	0,05
	Adult	0,008	0,008	0,07	0,05	0,008	0,01	0	0,008
64	Car Evaluation	0,79	0,77	0,8	0,78	**0,7**	0,6	0,71	0,61
	Bank Marketing	0,01	0,02	0,01	0,02	0,01	0,02	0,01	0,02
	Mushroom	0,98	0,99	0,99	0,99	**0,95**	0,95	0,95	0,95
	Nursery	0,01	0,02	0,01	0,02	0,01	0,01	0,01	0,01
	Adult	0,004	0,004	0,005	0,007	0,008	0,008	0,007	0,01

128	**Car Evaluation**	0,8	0,77	0,81	0,79	**0,73**	0,69	0,73	0,68
	Bank Marketing	0,88	0,78	0,89	0,83	0,88	0,78	0,89	0,83
	Mushroom	0,98	0,99	0,99	0,99	**0,91**	0,91	0,91	0,91
	Nursery	0,65	0,6	0,65	0,6	0,75	0,74	0,75	0,75
	Adult	0,004	0,007	0,005	0,005	0,004	0,03	0,005	0,008

4. SONUÇLAR

Bu tez çalışmasında mobil uygulamalar, nesnelerin interneti, akıllı cihazlar gibi istemci sunucu olarak birlikte çalışan sistemlerde, veriden habersiz ve artırımlı sınıflandırmaya dayalı birliktelik kuralları çıkarma modeli önerilmiştir. Önerilen modelin eğitiminde federe öğrenme mimarisi kullanılmıştır. Federe öğrenme mimarisi, istemci sunucu sistemlerinde veri mahremiyetini koruyarak model eğiten güncel bir öğrenme teknolojisidir. Bu özelliği ile merkezde veri toplayarak model eğiten klasik öğrenmelerden ayrılmaktadır. Model eğitiminde kişisel verileri koruduğu için GDPR ve KVKK gibi kanun ve yönetmeliklere uygun düşmektedir. Ayrıca bu öğrenme yöntemi istemciler ile sunucu arasındaki ağ trafiğini azaltmakta, enerji ihtiyacını düşürmekte ve güncel modeller ortaya çıkartmaktadır.

Gittikçe büyüyen ve çeşitliliği artan veriler için hem veri mahremiyetini koruyan hem de verinin üretildiği yerde işlenmesini sağlayan bu yenilikçi teknolojinin geliştirilip bu alanda farklı çalışmalar yapılmasına ihtiyaç duyulmaktadır. Tez çalışması kapsamında bu alanda bir algoritma geliştirilmiştir. Algoritma Veriden Habersiz İlişkili Kurallara Dayalı Sınıflandırma (Data Unaware Classification Based on Association, du-CBA) olarak adlandırılmıştır. du-CBA ilişkisel sınıflandırma yönteminin CBA algoritması üzerine çalışılarak geliştirilmiş yeni bir algoritmadır. Federe öğrenme mimarisi ile ilişkisel sınıflandırma alanında daha önce çalışma yapılmamıştır. Bu çalışma için motivasyonumuz ilişkisel sınıflandırma ile federe öğrenme mimarisi alanında çalışma yapılmamış olması ve bu alanda ki eksikliği bir parçada olsa kapatmaktır.

Çalışma kapsamında federe öğrenme ile klasik öğrenme mimarilerini karşılaştırıp başarılarını ölçmek için bir simülasyon ortamı oluşturulmuştur. Simülasyon ortamında iki farklı şekilde model eğitimi gerçekleştirilmiştir. Elde edilen modeller performans ölçütlerine ve eğitim sürelerine göre kıyaslanmıştır. du-CBA ve ilişkisel sınıflandırma algoritmalarından biri olan CBA kullanılarak, UCI veri havuzundan alınan beş farklı veri setleri ile modeller eğitilmiştir. Deneysel sonuçlar CBA ile karşılaştırıldığında, ayrı ayrı her bir veri seti için du-CBA ile model eğitim süresinin yaklaşık olarak %70 oranında azaldığını ve neredeyse aynı doğruluğu elde ettiğini göstermiştir. Yani istemci sunucu gibi birlikte çalışan sistemlerde makine öğrenmesi yöntemleri uygulanması durumunda doğruluğundan ödün vermeden ama daha hızlı sonuç üreten bir model eğitimi sağlanmaktadır. Ayrıca du-CBA ile eğitilen model gerçek zamanda sonuç üretmekte ve öğrenmeye devam etmektedir. Bu sebeple güncel kalmakta, artırımlı öğrenmeye de örnek olmaktadır.

Sonuçlara göre federe öğrenme ile eğitilen model diğer modele kıyasla daha kısa eğitim süresine sahip olduğu için, federe öğrenme mimarisini kullanacak cihazlarda enerji ihtiyacının düştüğü bilgisi desteklenmektedir. Ayrıca uçlardan veri yerine eğitilmiş model merkeze gönderilmiştir. Böylece veri mahremiyeti sağlanmıştır. Ağ trafiği de kayda değer şekilde azalmıştır.

Geliştirilen algoritmada birliktelik kuralları çıkartmak için Apriori algoritması kullanılmıştır. İlerleyen çalışmalarda diğer birliktelik kuralları algoritmaları kullanılarak duCBA'nın performansı değerlendirilecektir. Federe öğrenmesi mimarisi kapsamında veri güvenliğini sağlamak adına homomorfik şifreleme yöntemi kullanılarak çalışma yapılması hedeflenmektedir.

Ayrıca federe öğrenme yöntemi gerçek hayatta uygulandığında, kullanılan sistemlerin tam koordine edilememesinden kaynaklı bağlantı sorunları, modellerin güncelleme eksiklikleri, eğitim sürelerinin ve sürümlerinin farklı olması bu teknolojinin eksikliklerini ve zorluklarını göstermektedir. İlerleyen çalışmalarda federe öğrenmesi uygulanmasında meydana gelen bu sorunların çözümü için yeni algoritma ve yaklaşımların geliştirilmesi hedeflenmektedir.

KAYNAKLAR

1. Uygunoğlu, T. and İ.B. Topçu, Nesnelerin İnternetinin (Iot) İnşaat Mühendisliğindeki Rolü: Rfıd Uygulamaları. International Journal of 3D Printing Technologies and Digital Industry, 2020. 4(3): P. 270-277.
2. Yıldız, A., Endüstri 4.0 ve Akıllı Fabrikalar. Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 2018. 22(2): P. 546-556.
3. Lee, J., B. Bagheri, and H.-A. Kao, A Cyber-Physical Systems Architecture for İndustry 4.0-Based Manufacturing Systems. Manufacturing Letters, 2015. 3: P. 18-23.
4. Özhan, E., Güvenlik Duvarı Günlüklerinin Makine Öğrenmesi Yöntemleri ile Analizi ve Bir Model Çıkartılması. 2013.
5. Kavza, U., Veri Madenciliğinde Mahremiyetin Sağlanması. 2010, Mühendislik ve Fen Bilimleri Enstitüsü.
6. Büyüknacar, Y.C.-Y., Federe Öğrenme ve Veri Mahremiyeti.
7. Süzen, A.A. and K. Kayaalp, Büyük Verilerde Gizlilik Tabanlı Yaklaşım: Federe Öğrenme. International Journal of 3D Printing Technologies and Digital Industry. 3(3): P. 297-304.
8. Provost, F. and T. Fawcett, Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking. 2013: " O'reilly Media, Inc.".
9. Aydın, S., Veri Madenciliği ve Anadolu Üniversitesi Uzaktan Eğitim Sisteminde Bir Uygulama. 2015, Anadolu University (Turkey).
10. Alpaydin, E., Introduction to Machine Learning. 2020: Mıt Press.
11. Onan, A. and S. Korukoğlu, Makine Öğrenmesi Yöntemlerinin Görüş Madenciliğinde Kullanılması Üzerine Bir Literatür Araştırması. Pamukkale University Journal of Engineering Sciences, 2016. 22(2).
12. Ozan, C., İyileştirilmiş Pekiştirmeli Öğrenme Yöntemi ve Dinamik Yükleme ile Kentiçi Ulaşım Ağlarının Tasarımı. 2012.
13. Gökçay, B. and B. Arda, Kişisel Sağlık Verilerinin Korunması Kapsamında Sağlık Araştırmalarında Etik Bakış. Türk Kardiyoloji Derneği Arşivi, 2019. 47(3): P. 218-227.
14. Li, Q., Et Al., A Survey On Federated Learning Systems: Vision, Hype And Reality for Data Privacy and Protection. Ieee Transactions on Knowledge and Data Engineering, 2021.
15. Jiang, J.C., Et Al., Federated Learning in Smart City Sensing: Challenges and Opportunities. Sensors, 2020. 20(21): P. 6230.

16. Kairouz, P., Et Al., Advances and Open Problems in Federated Learning. Foundations and Trends® in Machine Learning, 2021. 14(1–2): P. 1-210.
17. Ma, C., Et Al., On Safeguarding Privacy and Security in The Framework of Federated Learning. Ieee Network, 2020. 34(4): P. 242-248.
18. Liu, Y., Et Al., A Systematic Literature Review on Federated Learning: From A Model Quality Perspective. Arxiv Preprint Arxiv:2012.01973, 2020.
19. Yang, Q., Et Al., Federated Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2019. 13(3): P. 1-207.
20. Gepperth, A. and B. Hammer. Incremental Learning Algorithms and Applications. In European Symposium on Artificial Neural Networks (Esann). 2016.
21. Hu, C., Et Al., A Novel Random Forests Based Class Incremental Learning Method for Activity Recognition. Pattern Recognition, 2018. 78: P. 277-290.
22. Bifet, A. and R. Kirkby, Data Stream Mining A Practical Approach. 2009.
23. Liu, B., W. Hsu, and Y. Ma. Integrating Classification and Association Rule Mining. İn Kdd. 1998.
24. Arslan, A.K., Et Al., A Novel Interpretable Web-Based Tool on The Associative Classification Methods: An Application on Breast Cancer Dataset. The Journal of Cognitive Systems, 2020. 5(1): P. 33-40.
25. Azmi, M., G.C. Runger, and A. Berrado, Interpretable Regularized Class Association Rules Algorithm for Classification In A Categorical Data Space. Information Sciences, 2019. 483: P. 313-331.
26. Thabtah, F., P. Cowling, and Y. Peng. Mcar: Multi-Class Classification Based on Association Rule. In The 3rd Acs/IEEE International Conference Oncomputer Systems and Applications, 2005. 2005. Ieee.
27. Yin, X. and J. Han. Cpar: Classification Based On Predictive Association Rules. In Proceedings of The 2003 Sıam International Conference On Data Mining. 2003. Sıam.
28. Yoon, Y. and G.G. Lee. Practical Application Of Associative Classifier for Document Classification. In Asia Information Retrieval Symposium. 2005. Springer.
29. Antonie, M.-L., O.R. Zaiane, and A. Coman. Associative Classifiers for Medical İmages. İn Pacific-Asia Conference On Knowledge Discovery And Data Mining. 2002. Springer.
30. Li, W., J. Han, and J. Pei. Cmar: Accurate and Efficient Classification Based on Multiple Class-Association Rules. In Proceedings 2001 IEEE International Conference On Data Mining. 2001. IEEE.

31. Alwidian, J., B. Hammo, and N. Obeid. Enhanced CBA Algorithm Based on Apriori Optimization and Statistical Ranking Measure. In Proceeding of 28th International Business Information Management Association (Ibıma) Conference on Vision. 2016.
32. Alwidian, J., B. Hammo, and N. Obeid, FCBA: Fast Classification Based on Association Rules Algorithm. International Journal of Computer Science and Network Security (Ijcsns), 2016. 16(12): P. 117.
33. Alwidian, J., B.H. Hammo, and N. Obeid, WCBA: Weighted Classification Based on Association Rules Algorithm for Breast Cancer Disease. Applied Soft Computing, 2018. 62: P. 536-549.
34. Alnababteh, M.H., Et Al., Associative Classification Based on Incremental Mining (ACIM). International Journal of Computer Theory and Engineering, 2014. 6(2): P. 135.
35. Al-Fayoumi, M.A., Enhanced Associative Classification Based on Incremental Mining Algorithm (E-ACIM). International Journal of Computer Science Issues (Ijcsı), 2015. 12(1): P. 124.
36. Bonawitz, K., Et Al., Towards Federated Learning At Scale: System Design. Proceedings of Machine Learning and Systems, 2019. 1: P. 374-388.
37. Hard, A., Et Al., Federated Learning for Mobile Keyboard Prediction. Arxiv Preprint Arxiv:1811.03604, 2018.
38. Hartmann, F. Federated Learning for Firefox. 27.08.2018 [Cited 2022 14.04.2022]; Available From: https://florian.github.io/federated-learning-firefox/
39. Jansson, M. and M. Axelsson, Federated Learning Used to Detect Credit Card Fraud. Lu-Cs-Ex, 2020.
40. Yuan, B., S. Ge, and W. Xing, A Federated Learning Framework for Healthcare IoT Devices. Arxiv Preprint Arxiv:2005.05083, 2020.
41. Kumar, R., Et Al., Blockchain-Federated-Learning and Deep Learning Models for Covid-19 Detection Using Ct Imaging. IEEE Sensors Journal, 2021. 21(14): P. 16301-16314.
[42. Yu, B., Et Al., A Survey on Federated Learning in Data Mining. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2022. 12(1): P. E1443.
43. Zheng, Z., Et Al., Applications of Federated Learning in Smart Cities: Recent Advances, Taxonomy, and Open Challenges. Connection Science, 2022. 34(1): P. 1-28.
44. Repository, U.M.L. Car Evaluation Data Set. [Cited 23.04.2022; Available From: https://archive.ics.uci.edu/ml/datasets/car+evaluation

45. Repository, U.M.L. Bank Marketing Data Set. [Cited 23.04.2022; Available From: https://archive.ics.uci.edu/ml/datasets/bank+marketing
46. Repository, U.M.L. Mushroom Data Set. 23.04.2022]; Available From: https://archive.ics.uci.edu/ml/datasets/mushroom
47. Repository, U.M.L. Nursery Data Set. 23.04.2022]; Available From: https://archive.ics.uci.edu/ml/datasets/nursery
48. Repository, U.M.L. Adult Data Set. 23.04.2022]; Available From: https://archive.ics.uci.edu/ml/datasets/adult
49. Agrawal, R. and R. Srikant. Fast Algorithms For Mining Association Rules. In Proc. 20th Int. Conf. Very Large Data Bases, Vldb. 1994. Citeseer.
50. Agrawal, R., T. Imieliński, and A. Swami. Mining Association Rules Between Sets of Items in Large Databases. In Proceedings of The 1993 Acm Sıgmod International Conference on Management Of Data. 1993.
51. Pycharm Ide. 07.06.2022]; Available From: https://www.jetbrains.com/pycharm/
52. Pyarc. 07.06.2022]; Available From: https://pypi.org/project/pyarc/
53. Abdelhamid, N., Multi-Label Rules for Phishing Classification. Applied Computing and Informatics, 2015. 11(1): P. 29-46.
54. Moh'd Iqbal, A., W.E. Hadi, and J. Alwedyan, Detecting Phishing Websites Using Associative Classification. Journal of Information Engineering and Applications, Voı, 2013. 3.
55. Thabtah, F., Et Al., Prediction Phase in Associative Classification Mining. International Journal of Software Engineering and Knowledge Engineering, 2011. 21(06): P. 855-876.
56. Badem, H., Parkinson Hastaliğinin Ses Sinyalleri Üzerinden Makine Öğrenmesi Teknikleri ile Tanimlanmasi. Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi, 2019. 8(2): P. 630-637.
57. Hossin, M. and M.N. Sulaiman, A Review on Evaluation Metrics for Data Classification Evaluations. International Journal of Data Mining & Knowledge Management Process, 2015. 5(2): P. 1.

ÖZGEÇMİŞ

Adı Soyadı **:** Büşra BÜYÜKTANIR

Öğrenim Durumu

Derece	Bölüm/Program	Üniversite/Lise	Mezuniyet Yılı
Lise	Anadolu Lisesi	Hüseyin Özbuğday Anadolu Lisesi	Haziran 2010
Lisans	Bilgisayar Mühendisliği	Kırıkkale Üniversitesi	Haziran 2014

İş Deneyimi

Yıl	Firma/Kurum	Görevi
2016	Uşak Sinüs Eğitim Kurumları	Yazılım Eğitmeni
2019	Altınbaş Üniversitesi	Araştırma Görevlisi
2021	Marmara Üniversitesi	Araştırma Görevlisi

Bilimsel Eserler:

1. Büyüktanır T., Büyüktanır B. (2017) Nesnelerin İnterneti. Kodlab Yayıncılık

2. Pınar, M., Büyüktanır, B., Emanet, Ş., & Doğan, B. (2020). Yazılım Projelerinde Fazla Mesainin Proje Ekibi ve Projenin Yönetimine Etkisi. *International Journal of Advances in Engineering and Pure Sciences*, *32*(4), 420-429.

3. Dalip F., Çakar M., Büyüktanır B, Doğan B. (2021, Mayıs). Effects of Covid-19 Pandemic on Distance Workers. II. International Covid-19 and Current Issues Congress.

4. Özdemir, M., Yıldız, K., & Büyüktanır, B. (2022) Housing Price Estimation with Deep Learning: A Case Study of Sakarya Turkey. Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi, 9(1), 138-151.