MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

FEDERE ÖĞRENME MİMARİSİNDE VERİDEN HABERSİZ İLİŞKİSEL SINIFLANDIRMA YÖNTEMİ İLE EĞİTİLMİŞ MODELLER İÇİN UNUTMA: YENİ BİR MAKİNE UNUTMASI YAKLAŞIMI

BÜŞRA BÜYÜKTANIR

DOKTORA TEZİ

Teknoloji Fakültesi Bilgisayar Mühendisliği Bölümü

DANIŞMAN

Prof. Dr. Kazım YILDIZ

EŞ-DANIŞMAN

Dr. Gözde KARATAŞ BAYDOĞMUŞ

İSTANBUL, 2026

MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES IN PURE AND APPLIED SCIENCES

FORGETTING FOR MODELS TRAINED WITH A DATA-UNAWARE ASSOCIATION CLASSIFICATION METHOD IN FEDERATED LEARNING ARCHITECTURE: A NOVEL MACHINE UNLEARNING APPROACH

BÜŞRA BÜYÜKTANIR

Ph.D. THESIS
Department of Computer Engineering, Faculty of Technology

THESIS SUPERVISOR
Prof. Dr. Kazım YILDIZ

THESIS CO-SUPERVISOR
PhD. Gözde KARATAŞ BAYDOĞMUŞ

İSTANBUL, 2026

ÖNSÖZ

Öncelikle, mülkün gerçek sahibi olan Allah'a sonsuz hamd ve şükürlerimi sunarım.

Doktora eğitimim ve tez çalışmam süresince bilgi, deneyim ve kıymetli yönlendirmeleriyle her zaman destek olan değerli danışmanım Prof. Dr. Kazım YILDIZ'a ve eş danışmanım Dr. Öğr. Üyesi Gözde Karataş Baydoğmuş'a içten teşekkürlerimi sunarım. Ayrıca lisans, yüksek lisans ve doktora dönemim boyunca bilgi ve deneyimlerinden faydalandığım tüm kıymetli hocalarıma katkılarından dolayı teşekkür ederim.

Hayatımın her alanında olduğu gibi doktora eğitimim ve tez çalışmam boyunca da desteğini, sabrını, sevgisini ve anlayışını hiçbir zaman eksik etmeyen kıymetli eşim Tolga BÜYÜKTANIR'a; bugünlere gelmemde büyük emeği olan değerli annem Fatma TEKE'ye ve babam Halil ÖZER'e en derin teşekkürlerimi sunarım.

Her zaman yanımda olan, desteği ve sevgisiyle bana güç veren kıymetli görümcem ve kızlarımın biricik halası Ülkü BÜYÜKTANIR'a; manevi desteklerini hiçbir zaman eksik etmeyen canım kardeşlerim Hatice Nur ÖZER'e, Halil İbrahim ÖZER'e ve Oğuz Han ÖZER'e; ayrıca duaları ve destekleriyle yanımda olan tüm kıymetli dostlarıma teşekkür ederim.

Bununla birlikte, bu uzun ve zorlu süreçte kendime de teşekkür etmek istiyorum. Bir anne, bir eş, bir öğrenci, çalışan bir kadın ve Müslüman bir hanım olarak karşılaştığım tüm zorluklara rağmen vazgeçmeden mücadele ettiğim, sabırla çalıştığım ve inancımı koruyarak bu yolu tamamladığım için kendime şükran duyuyorum.

Son olarak, bu çalışmayı varlıklarıyla hayatıma anlam, gülüşleriyle bana güç katan güzel kızlarım Zeynep Şeyda'ya ve Eslem Esra'ya armağan ediyorum.

MAYIS, 2026 **Büşra BÜYÜKTANIR**

İÇİNDEKİLER

ÖZET iii
ABSTRACT v
YENİLİK BEYANI vii
SEMBOLLER viii
KISALTMALAR x
ŞEKİL LİSTESİ xiii
TABLO LİSTESİ xiv
1. GİRİŞ 1
1.1. Problem Tanımı 4
1.2. Çalışmanın Arka Planı ve Çözüm Yaklaşımı 5
1.3. Çalışmanın Amacı ve Motivasyonu 6
1.4. Literatüre Temel Katkılar 6
1.5. Tezin Organizasyonu 7
2. TEORİK ALTYAPI VE İLGİLİ ÇALIŞMALAR 8
2.1. Makine Unutması (Machine Unlearning) 8
2.2. Federe Öğrenme (Federated Learning) 29
2.3. Federe Öğrenme Ortamında Makine Unutması (Federated Unlearning) 40
2.4. İlişkisel Sınıflandırma (Associative Classification) 46
3. YÖNTEM 54
3.1. Sisteme Genel Bakış 54
3.2. FedCBA Algoritmasının Tasarımı 56
3.2.1. Tanım ve mimari temeller 56
3.2.2. Kural oluşturma ve birleştirme süreçleri 57
3.2.3. Matematiksel formüller 57
3.2.4. FedCBA sözde kodu 59
3.3. FedFORGET: Federe Mimaride Makine Unutması Yaklaşımı 61
3.3.1. Tanım ve mimari temeller 61
3.3.2. Kural çıkartma süreçleri 61
3.3.3. Matematiksel formüller 62
3.3.4. FedFORGET sözde kod 63
4. DENEYSEL ÇALIŞMALAR VE BULGULAR 66
4.1. Veri Setleri 67
4.2. Değerlendirme Kriterleri 72
4.3. FedCBA Performans Analizi ve Sonuçları 73

4.3.1. Deney ortamı 74
4.3.2. Deneysel bulgular ve performans karşılaştırması 76
4.4. FedFORGET Unutma Başarımı ve Sonuçları 79
4.4.1. Deney ortamı 79
4.4.2. Deneysel bulgular ve performans karşılaştırması 82
4.5. Raspberry Pi İstemcileri ile Gerçekleştirilen Federe Unutma Performans Analizi 93
4.5.1. Deney ortamı 93
4.5.2. Deneysel Bulgular 96
4.6. Bulut Ortamında Federe Unutma Performans Analizi 98
4.6.1. Deney ortamı 100
4.6.2. Deneysel Bulgular 102
5. TARTIŞMA VE SONUÇ 106
5.1. Elde Edilen Bulguların Değerlendirilmesi ve Tartışılması 106
5.2. Sistemin Avantajları ve Kısıtları 107
5.3. Gelecek Çalışmalar için Öneriler 107

ÖZET

Federe öğrenme mimarileri, kullanıcı verilerini merkezi sunucuya taşımadan model eğitimi gerçekleştirebildiği için veri gizliliği açısından önemli avantajlar sunmaktadır. Ancak bu mimarilerde, kullanıcıların sistemden ayrılması veya verilerinin silinmesini talep etmesi durumunda, ilgili verinin model üzerindeki etkisinin tamamen kaldırılması önemli bir problem oluşturmaktadır. Geleneksel yeniden eğitim tabanlı yaklaşımlar yüksek hesaplama maliyeti, iletişim yükü ve zaman gereksinimi nedeniyle büyük ölçekli federe sistemlerde uygulanabilir değildir.

Bu tez çalışmasında, federe öğrenme mimarileri için ilişkisel sınıflandırma tabanlı yeni bir federe birleştirme algoritması olan Federated Classification Based on Associations (FedCBA) ve bu yapı üzerine geliştirilen veriden habersiz federe unutma yaklaşımı olan Federated Forgetting (FedFORGET) önerilmiştir. Önerilen mimari, istemcilerden elde edilen yerel Classification Based on Associations (CBA) ile eğitilmiş modellerin merkezi sunucuda kural tabanlı olarak birleştirilmesini sağlamakta ve unutma işlemlerini orijinal verilere erişim gerektirmeden gerçekleştirebilmektedir. Ayrıca geliştirilen yaklaşım, kural tabanlı yapısı sayesinde açıklanabilir ve doğrulanabilir bir unutma mekanizması sunmaktadır.

Çalışma kapsamında önerilen yöntemler; IBM İnsan Kaynakları Analitiği, Adult, Mushroom, Nursery ve Cars Evaluation veri setleri kullanılarak bağımsız ve özdeş dağılım (Independent and Identically Distributed - IID) ve bağımsız ve özdeş olmayan dağılım (Non-Independent and Non-Identically Distributed - Non-IID) veri yapıları altında test edilmiştir. Deneysel çalışmalar yalnızca simülasyon ortamında gerçekleştirilmemiş; Google Cloud tabanlı uygulama programlama arayüzü (Application Programming Interface - API) destekli dağıtık mimari ve Raspberry Pi tabanlı fiziksel istemci ortamlarında da uygulanmıştır. Böylece önerilen yaklaşımın gerçek dünya koşullarındaki uygulanabilirliği analiz edilmiştir.

Elde edilen deneysel sonuçlar, FedCBA yaklaşımının farklı veri dağılımlarında kararlı performans sonuçları üretebildiğini göstermiştir. Özellikle Non-IID veri yapısında dahi global model performansının korunabildiği ve istemcilerden gelen yerel modellerin başarılı şekilde birleştirilebildiği gözlemlenmiştir. Gerçekleştirilen performans analizlerinde, FedCBA yaklaşımının diğer federe birleştirme yöntemleri ile benzer doğruluk seviyelerinde sonuçlar üretebildiği gözlemlenmiştir.

FedFORGET sonuçları incelendiğinde, unutma işlemi sonrasında hedef istemci doğruluklarında belirgin düşüşler meydana gelirken global model performansının büyük ölçüde korunabildiği gözlemlenmiştir. Elde edilen sonuçlar, geliştirilen yöntemin hedef istemciye ait bilgiyi modelden başarıyla çıkarabildiğini göstermektedir. Ayrıca unutma işleminin hem yerel istemci verileri hem de global test verileri üzerinde etkili biçimde gerçekleştirildiği görülmüştür. Yeniden eğitim tabanlı yöntemlerle karşılaştırıldığında ise FedFORGET yaklaşımının önemli ölçüde zaman avantajı sağladığı ve modeli baştan eğitmeye ihtiyaç duymadan unutma işlemini gerçekleştirebildiği belirlenmiştir.

Sonuç olarak bu tez çalışması, federe öğrenme mimarilerinde veri gizliliğini koruyarak çalışan, veriden habersiz ve istemci düzeyinde unutma gerçekleştirebilen yeni bir makine unutması yaklaşımı sunmaktadır. Ayrıca geliştirilen FedCBA ve FedFORGET yöntemlerinin yalnızca teorik değil, gerçek dağıtık sistem ortamlarında da uygulanabilir olduğu deneysel olarak gösterilmiş ve literatürdeki yeniden eğitim tabanlı unutma yöntemlerine alternatif bir çözüm sunduğu ortaya konulmuştur.

ABSTRACT

Federated learning architectures provide significant advantages in terms of data privacy, as model training can be performed without transferring user data to a central server. However, in such architectures, completely removing the influence of a client's data from the trained model becomes a critical challenge when users leave the system or request the deletion of their data. Traditional retraining-based approaches are not practical for large-scale federated systems due to their high computational cost, communication overhead, and time requirements.

In this dissertation, a novel rule-based federated aggregation algorithm, namely Federated Classification Based on Associations (FedCBA), and a data-unaware federated unlearning approach built upon this architecture, namely Federated Forgetting (FedFORGET), are proposed for federated learning environments. The proposed architecture enables the rule-based aggregation of locally trained Classification Based on Associations (CBA) models at the central server and performs unlearning operations without requiring access to the original training data. Furthermore, owing to its rule-based structure, the proposed approach provides an interpretable and verifiable unlearning mechanism.

The proposed methods were evaluated using the IBM HR Analytics, Adult, Mushroom, Nursery, and Cars Evaluation datasets under both Independent and Identically Distributed (IID) and Non-Independent and Non-Identically Distributed (Non-IID) data distributions. Experimental studies were not limited to simulation environments; the proposed architecture was also implemented within a Google Cloud-based Application Programming Interface (API)-supported distributed system and Raspberry Pi-based physical client environments. In this way, the applicability of the proposed approach under real-world distributed system conditions was comprehensively analyzed.

The experimental results demonstrated that the FedCBA approach was capable of producing stable performance across different data distributions. In particular, even under Non-IID data settings, the global model performance was largely preserved, and the local models received from clients were successfully aggregated. Performance analyses further revealed that the FedCBA approach achieved comparable accuracy levels to other federated aggregation methods.

The results obtained from the FedFORGET experiments showed that while the target client accuracies significantly decreased after the unlearning process, the overall global model performance was largely maintained. These findings indicate that the proposed method

successfully removes the contribution of the target client from the global model. In addition, the unlearning process was shown to operate effectively on both local client datasets and global test datasets. Compared with retraining-based approaches, the FedFORGET method provided a considerable time advantage and was able to perform unlearning without retraining the model from scratch.

Consequently, this dissertation proposes a novel machine unlearning approach for federated learning architectures that preserves data privacy, operates in a data-unaware manner, and performs client-level unlearning. Furthermore, the experimental results demonstrated that the proposed FedCBA and FedFORGET methods are not only theoretically valid but also practically applicable in real distributed system environments, thereby offering an alternative solution to conventional retraining-based unlearning methods in the literature.

YENİLİK BEYANI

Bu tez çalışması, federe öğrenme mimarilerinde veri gizliliğini koruyarak makine unutmasını gerçekleştiren özgün bir yaklaşım sunmaktadır. Literatürdeki mevcut makine unutması ve federe öğrenme çalışmalarından farklı olarak, bu tezin literatüre sunduğu temel yenilikler ve özgün katkılar aşağıda özetlenmiştir:

Federe Ortamda Birliktelik Kurallarına Dayalı Sınıflandırma (Federated Classification Based on Associations - FedCBA): Geleneksel federe öğrenme yaklaşımlarının ağırlık veya gradyan tabanlı birleştirme yöntemlerinin aksine, bu çalışmada kural tabanlı ilişkisel sınıflandırma algoritmaları federe mimariye entegre edilmiştir. İstemcilerden gelen ilişki kurallarının merkezi sunucuda etkin bir şekilde birleştirilmesini sağlayan FedCBA algoritması geliştirilerek literatüre yeni bir yöntem kazandırmıştır [1].

Veriden Habersiz Federe Makine Unutması (Federated Forgetting - FedFORGET): Makine unutması literatüründeki en büyük zorluklardan biri, unutma işlemi sırasında modelin yeniden eğitilmesi veya geçmiş verilere ihtiyaç duyulmasıdır. Bu tez kapsamında geliştirilen FedFORGET algoritması, tamamen veriden habersiz bir yapı sunarak, silinmesi talep edilen verinin etkilerini orijinal veri setine ihtiyaç duymadan, doğrudan merkezi sunucu üzerinde global kural tabanından kaldırabilen yenilikçi bir unutma mekanizması önermektedir.

Bulut ve Dağıtık Mimari Uygulaması: Geliştirilen yöntemin performansı, yerel simülasyon ortamlarına ek olarak Google Cloud altyapısı üzerinde API haberleşmesiyle ve Raspberry Pi donanımları kullanılarak uçtan uca test edilmiştir. Bu deneysel kurgu ile, önerilen yöntemin gerçek dünya senaryolarında uygulanabilirliği analiz edilmiştir.

SEMBOLLER

$\mathbf{D_{(i)}}$	:i. istemciye ait yerel veri setini ifade eder.
$\boldsymbol{n}$	:Sistemde yer alan toplam istemci sayısını gösterir.
$\boldsymbol{d}$	:Bir kuralın geçerli kabul edilebilmesi için gerekli minimum destek değeridir. (Support threshold)
$\boldsymbol{g}$	:Bir kuralın güvenilir kabul edilebilmesi için gerekli minimum güven değeridir. (Confidence threshold)
$\boldsymbol{M^{(i)}_{lokal}}$	: i. istemcide CBA algoritması ile eğitilen yerel modeldir.
$\boldsymbol{M_{global}}$	:Tüm istemcilerden gelen kuralların birleştirilmesi sonucu elde edilen global modeldir.
$\boldsymbol{N_i}$	:i. istemcinin veri setindeki toplam örnek sayısını ifade eder.
Toplam_Ornek	**:**Tüm istemcilerdeki veri örneklerinin toplamını gösterir.
$\boldsymbol{r_{yerel}}$	**:**Yerel modelde bulunan bir birliktelik kuralıdır.
$\boldsymbol{r_{global}}$	:Global modelde bulunan bir birliktelik kuralıdır.
lhs	**:**Kuralın öncül kısmını ifade eder. (Left-Hand Side)
rhs	**:**Kuralın sonuç kısmını ifade eder. (Right-Hand Side)
Frekans (X, Y)	**:**X ve Y'nin birlikte görülme sıklığını ifade eder.
Frekans (X)	:X'in tek başına görülme sıklığını ifade eder.
t	:Global modeldeki kümülatif katkıyı temsil eder.
a	:Sistemden ayrılan istemcinin katkısını temsil eder.
R_global	:FedCBA ile elde edilen global modele ait kural seti.
R_ayrılan	:Sistemden ayrılmak isteyen istemciye ait yerel kural seti.
R_yeni	:FedFORGET işlemi sonrasında elde edilen güncellenmiş global kural seti.
N_toplam	:Global modelin oluşturulmasında kullanılan toplam örneklem sayısı.
N_ayrılan	:Sistemden ayrılan istemciye ait yerel örneklem sayısı.

N_yeni :Unutma işlemi sonrasında güncellenmiş toplam örneklem sayısı.

r_ayrılan :Ayrılan istemcinin yerel modelinde yer alan herhangi bir kural.

destek_global :Global modeldeki ilgili kuralın destek değeri.

destek_ayrılan :Ayrılan istemcideki ilgili kuralın destek değeri.

güven_global :Global modeldeki ilgili kuralın güven değeri.

güven_ayrılan :Ayrılan istemcideki ilgili kuralın güven değeri.

yeni_destek :Ayrılan istemcinin katkısı çıkarıldıktan sonra hesaplanan destek değeri.

yeni_güven :Ayrılan istemcinin katkısı çıkarıldıktan sonra hesaplanan güven değeri.

D_i^t **:**i. istemcinin t. turdaki yerel veri kümesi.

M_i^t :i. istemcinin t. tur sonunda oluşturduğu yerel CBA modeli.

G^t **:**t. tur sonunda elde edilen global model.

G^{-k} :k. istemcisinin katkısı çıkarıldıktan sonra elde edilen unutmuş global. model.

T :Toplam federe tur sayısı.

KISALTMALAR

ACIM : Artırımlı Madenciliğe Dayalı Birliktelik Tabanlı Sınıflandırma, Associative Classifier with Instance Matching

API : Uygulama Programlama Arayüzü, Application Programming Interface

AR-GE: Araştırma ve Geliştirme

BFU : Bayesçi Federe Unutma, Bayesian Federated Unlearning

CAR : Sınıf Birliktelik Kuralları, Class Association Rules

CARMA : Sürekli Birliktelik Kuralı Madenciliği Algoritması, Continuous Association Rule Mining Algorithm

CBA : Birliktelik Kurallarına Dayalı Sınıflandırma, Classification Based on Associations

CCPA : Kaliforniya Tüketici Gizliliği Yasası, California Consumer Privacy Act

CHAC : Hibrit Birliktelik Tabanlı Sınıflandırıcı ile Sınıflandırma, Classification using Hybrid Association Classifier

CMAR : Çoklu Sınıf Birliktelik Kurallarına Dayalı Sınıflandırma, Classification Based on Multiple Class Association Rules

CPAR : Tahmin Edici Birliktelik Kurallarına Dayalı Sınıflandırma, Classification Based on Predictive Association Rules

CRAB : Zehirleme Saldırıları ve İhlallerden Sertifikalı Kurtarma, Certified Recovery from Poisoning Attacks and Breaches

DRA : Veri Yeniden Yapılandırma Saldırısı, Data Reconstruction Attack

du-CBA : Veriden Habersiz Birliktelik Kurallarına Dayalı Sınıflandırma, Data Unaware Classification Based on Association

E-ACIM : Artırımlı Madenciliğe Dayalı Geliştirilmiş Birliktelik Tabanlı Sınıflandırma, Enhanced Associative Classifier with Instance Matching

Eclat : Eşdeğer Sınıf Dönüşümü, Equivalence Class Transformation

FCBA : Birliktelik Kurallarına Dayalı Hızlı Sınıflandırma, Fast Classification Based on Associations

FedAvg : Federated Averaging

FedBE : Federated Bayesian Ensemble

FedCBA : Federated Classification Based on Associations

FedFast : Hızlı Federe Öğrenme, Federated Fast Learning

FedFORGET : Federated Forgetting

FedLU : Bilgi Grafikleri için Federe Öğrenme Tabanlı Unutma, Federated Learning-based Unlearning for Knowledge Graphs

FedMA : Federated Matched Averaging

FedNova : Federated Normalized Averaging

FedPer : Federated Personalization

FedProx : Federated Optimization with Proximal Term

FedRetrain: Federe Yeniden Eğitim, Federated Retraining

FL : Federe Öğrenme, Federated Learning

FP-Growth : Sık Örüntü Büyütme, Frequent Pattern Growth

FTL : Federe Transfer Öğrenme, Federated Transfer Learning

FU : Federe Unutma, Federated Unlearning

GAM : Genelleştirilmiş Toplamsal Modeller, Generalized Additive Models

GDPR : Avrupa Birliği Genel Veri Koruma Tüzüğü, General Data Protection Regulation

HFL : Yatay Federe Öğrenme, Horizontal Federated Learning

IBM: IBM İnsan Kaynakları Analitiği Çalışan Yıpranması/İşten Ayrılma ve Performans veri seti, IBM HR Analytics Employee Attrition & Performance dataset

IID : Bağımsız ve Özdeş Dağılım, Independent and Identically Distributed

IoT : Nesnelerin İnterneti, Internet of Things

İK : İnsan Kaynakları

KNN : K-En Yakın Komşu, K-Nearest Neighbors

KVKK : Kişisel Verilerin Korunması Kanunu

LSTM : Uzun Kısa Süreli Bellek, Long Short-Term Memory

MCAR : Birliktelik Kuralına Dayalı Çok Sınıflı Sınıflandırma, Multiclass Classification Based on Association Rule

MIA : Üyelik Çıkarım Saldırısı, Membership Inference Attack

MU : Makine Unutması, Machine Unlearning

msCBA : Çoklu Destekli CBA, Multiple Support CBA

Non-IID : Bağımsız ve Özdeş Olmayan Dağılım, Non-Independent and Non-Identically Distributed

OCD : Aday Belirleme, Off-line Candidate Determination

QuickDrop : Veri Damıtma Tabanlı Federe Unutma, Dataset Distillation-based Federated Unlearning

SIFU : Sıralı Bilgilendirilmiş Federe Unutma, Sequential Informed Federated Unlearning

SISA : Bölümlenmiş, İzole Edilmiş, Dilimlenmiş ve Birleştirilmiş, Sharded, Isolated, Sliced, Aggregated

SVM : Destek Vektör Makineleri, Support Vector Machines

TEE : Güvenilir Çalıştırma Ortamları, Trusted Execution Environments

VeriFi : Doğrulanabilir Federe Unutma, Verifiable Federated Unlearning

VFL : Dikey Federe Öğrenme, Vertical Federated Learning

WCBA : Ağırlıklı Birliktelik Kurallarına Dayalı Sınıflandırma, Weighted Classification Based on Associations

XAI : Açıklanabilir Yapay Zekâ, Explainable Artificial Intelligence

ŞEKİL LİSTESİ

Şekil 1.1. Merkezi, Dağıtık ve Federe öğrenme yaklaşımları arasındaki yapısal farklılıklar. 2
Şekil 1.2. Makine öğrenmesi modellerinde unutma verimliliği ile model doğruluğu arasındaki yapısal ödünleşim analizi [17]. 5
Şekil 2.1. Makine unutması sürecinin temel işleyişi. 9
Şekil 2.2. Makine Unutması: Teknikleri, Metrikleri ve Uygulama Alanları. 12
Şekil 2.3. "Machine unlearning" anahtar kelimesi kullanılarak Google Scholar veri tabanında yapılan arama sonuçlarının 2015–2025 yılları arasındaki yıllara göre dağılımı. 22
Şekil 2.4. Federe öğrenme mimarisi. 30
Şekil 2.5. Çapraz-silo federe öğrenme**.** 30
Şekil 2.6.Çapraz-cihaz federe öğrenme. 31
Şekil 2.7. Federe öğrenme ile model geliştirme süreci. 31
Şekil 2.8. IID ve Non-IID veri dağılımlarının istemciler üzerindeki karşılaştırmalı gösterimi. ...35
Şekil 2.9. Yatay federe öğrenme. 36
Şekil 2.10. Dikey federe öğrenme. 37
Şekil 2.11. Federe transfer öğrenme. 37
Şekil 2.12. "Federated learning" anahtar kelimesi kullanılarak Google Scholar veri tabanında yapılan arama sonuçlarının 2015–2025 yılları arasındaki yıllara göre dağılımı. 39
Şekil 2.13. Makine unutması ve yeniden eğitim süreçlerinin şematik gösterimi. 41
Şekil 2.14. Federe unutmanın; sınıf düzeyinde, örnek düzeyinde ve istemci düzeyinde iş akışları [17]. 41
Şekil 2.15. Federe unutma iş akış şeması [16]. 43
Şekil 2.16. "Federated unlearning" anahtar kelimesi kullanılarak Google Scholar veri tabanında yapılan arama sonuçlarının 2015–2025 yılları arasındaki yıllara göre dağılımı. 43
Şekil 2.17. İlişkisel sınıflandırma yöntemi akış diyagramı. 50
Şekil 3.1. Önerilen çerçevenin genel işleyişi: (a) FedCBA ile kural üretimi ve birleştirme, (b) İstemci tabanlı unutma talebinin başlatılması, (c) Sunucu tarafında FedFORGET unutma stratejisinin uygulanması. 55
Şekil 4.1. IBM İnsan Kaynakları Analitiği veri setindeki hedef değişkenin sınıflara göre dağılımı. 69
Şekil 4.2. Adult veri setindeki hedef değişkenin dengelenmiş sınıf dağılımı. 70
Şekil 4.3. Mushroom veri setindeki hedef değişkenin sınıflara göre dağılımı. 70
Şekil 4.4. Cars Evaluation veri setindeki hedef değişkenin sınıflara göre dağılımı. 71
Şekil 4.5. Nursery veri setindeki hedef değişkenin sınıflara göre dağılımı. 72
Şekil 4.6. IBM İnsan Kaynakları Analitiği veri setinin federe öğrenme mimarisindeki istemcilere departman bilgisine göre Non-IID dağılımı 75
Şekil 4.7. IBM İnsan Kaynakları Analitiği veri seti kullanılarak, farklı veri dağılımlarına sahip istemcilerde eğitilen yerel CBA modellerinin, FedCBA yaklaşımı ile merkezi sunucuda birleştirilerek global modelin oluşturulması süreci. 75
Şekil 4.8. FedCBA ile federe birleştirme ve FedFORGET ile istemci bazlı unutma süreci. 80
Şekil 4.9. Adult veri setinin federe öğrenme mimarisindeki istemcilere bölge bilgisine göre dağılımı. 81
Şekil 4.10. Test verileri için tahmin doğruluğu. 84
Şekil 4.11. Hedef veriler için tahmin doğruluğu. 86
Şekil 4.12. IBM veri setinin departman bazlı istemci dağılımı. 87
Şekil 4.13. Adult veri setinin bölge bazlı istemci dağılımı. 87
Şekil 4.14. Mushroom veri seti için farklı istemci sayılarında FedFORGET yönteminin performans sonuçları. 90

Şekil 4.15. Nursery veri seti için farklı istemci sayılarında FedFORGET yönteminin performans sonuçları. 90
Şekil 4.16. Cars Evaluation veri seti için farklı istemci sayılarında FedFORGET yönteminin performans sonuçları. 91
Şekil 4.17. Raspberry Pi tabanlı fiziksel istemciler, merkezi sunucu ve localhost arayüzünden oluşan FedFORGET sistem mimarisi.....93
Şekil 4.18. Deneysel çalışmalarda kullanılan Raspberry Pi tabanlı fiziksel istemci düzeneği. 94
Şekil 4.19. Sunucu bilgisayar üzerinde çalışan localhost tabanlı yönetim ve kontrol arayüzü.....95
Şekil 4.20. Google Cloud ortamında gerçekleştirilen FL ve FU mimarisinin genel yapısı. 99

TABLO LİSTESİ

Tablo 2.1. Makine unutması sürecinin avantaj ve dezavantajları. 10
Tablo 2.2. Kesin unutma ve yaklaşık unutma yöntemlerinin karşılaştırması [34]. 13
Tablo 2.3. Literatüre kazandırılmış unutma yöntemlerinin karşılaştırılması. 18
Tablo 2.4. Literatürde geliştirilen ve açık kaynak olarak sunulan makine unutması algoritmaları. 22
Tablo 2.5. Makine unutması çalışmalarının uygulama alanlarına göre sınıflandırılması. 25
Tablo 2.6. Federe öğrenme birleştirme algoritmaları. 33
Tablo 2.7. Federe öğrenmenin avantajları ve dezavantajları. 37
Tablo 2.8. Federe öğrenme çalışmalarının uygulama alanlarına göre sınıflandırılması. 39
Tablo 2.9. FU yöntemlerinin karşılaştırmalı özeti. 44
Tablo 2.10. Apriori, Eclat ve FP-Growth algoritmalarının karşılaştırması. 49
Tablo 2.11. 2025 yılında CBA algoritması ve ilişkisel sınıflandırma üzerine gerçekleştirilen güncel çalışmalar. 53
Tablo 4.1. IBM İnsan Kaynakları Analitiği veri setine ait temel özellikler. 68
Tablo 4.2. Adult veri setine ait temel özellikler. 69
Tablo 4.3. Mushroom veri setine ait temel özellikler. 70
Tablo 4.4. Cars Evaluation veri setine ait temel özellikler. 71
Tablo 4.5. Nursery veri setine ait temel özellikler. 72
Tablo 4.6. IBM İnsan Kaynakları Analitiği veri setinin federe mimari kapsamındaki veri dağılım parametreleri. 74
Tablo 4.7. Birliktelik kuralı algoritmalarının performans karşılaştırması. 77
Tablo 4.8. Dengelenmiş ve kısmen IID veri dağılımında federe birleştirme algoritmalarının performans karşılaştırması. 78
Tablo 4.9. Dengesiz ve Non-IID veri dağılımında federe birleştirme algoritmalarının performans karşılaştırması. 78
Tablo 4.10. Adult veri setinin federe mimari kapsamındaki veri dağılım parametreleri. 81
Tablo 4.11. Federe öğrenme ve unutma süreçleri için veri seti dağılım stratejileri. 82
Tablo 4.12. Test verileri üzerinde elde edilen performans sonuçları. 84
Tablo 4.13. Hedef veriler üzerinde elde edilen performans sonuçları. 85
Tablo 4.14. Federe öğrenme ve unutma sürelerinin karşılaştırılması (Saniye). 86
Tablo 4.15. Non-IID ortamda sırasıyla hedef istemciler unutulduktan sonra test verileri üzerinde elde edilen performans sonuçları. 88
Tablo 4.16. Non-IID ortamda sırasıyla hedef istemciler unutulduktan sonra hedef veriler üzerinde elde edilen performans sonuçları. 88
Tablo 4.17. Cars Evaluation veri seti kullanılarak global modelin oluşturulması ve istemci katkılarının analizi. 91
Tablo 4.18. Cars Evaluation veri setinde global model oluşturulduktan sonra İstemci 3'ün unutulması sonrası kural ve performans metriklerindeki değişim. 92
Tablo 4.19. Raspberry Pi ortamında gerçekleştirilen deneysel senaryoların özeti. 96
Tablo 4.20. Raspberry Pi ortamında IID ve Non-IID veri dağılımlarında yerel CBA modellerinin doğruluk sonuçları. 97
Tablo 4.21. Raspberry Pi ortamında FedCBA ve FedFORGET doğruluk sonuçları. 98
Tablo 4.22. Google Cloud ortamında gerçekleştirilen deneysel senaryoların özeti. 102
Tablo 4.23. İki turlu Non-IID veri dağılımı altında her istemci için ayrı ayrı gerçekleştirilen FedCBA ve FedFORGET performans sonuçları. 103
Tablo 4.24. İki turlu IID veri dağılımı altında her istemci için FedCBA ve İstemci 1 için FedFORGET performans sonuçları. 104

1. GİRİŞ

Günümüzde yapay zeka ve makine öğrenmesi teknolojileri; otonom araçlar, akıllı sağlık teşhis sistemleri, finansal risk analizleri, nesnelerin interneti (Internet of Things - IoT) ve siber güvenlik gibi veriye dayalı birçok kritik sektörde dönüştürücü bir rol oynamaktadır [2]. Bu teknolojiler ile eğitilen modellerin yüksek genelleme yeteneğine ulaşabilmesi, büyük ölçekli verilerin toplanmasını zorunlu kılmaktadır. Özellikle karmaşık yapıdaki öğrenme algoritmalarının eğitimi için büyük ölçekli verilerin düzenli olarak işlenmesi, kritik bir öneme sahiptir [3]. Bu durum, büyük veri depolama alanlarına ve yüksek kapasiteli donanımsal işlemcilere duyulan ihtiyacı da beraberinde getirmektedir. Tüm verilerin tek bir merkezi lokasyonda toplanmasının yarattığı yüksek donanım maliyetleri ve ağ darboğazları, makine öğrenmesi ve derin öğrenme algoritmalarının dağıtık donanım kaynakları üzerinde çalıştırılmasını zorunlu kılmıştır. Dağıtık öğrenme mimarilerinin sunduğu paralel hesaplama gücü ve esnek kaynak yönetimi sayesinde, yapay zeka modellerinin eğitim süreleri kısalmış ve çok daha ölçeklenebilir bir gelişim imkanı elde edilmiştir [4].

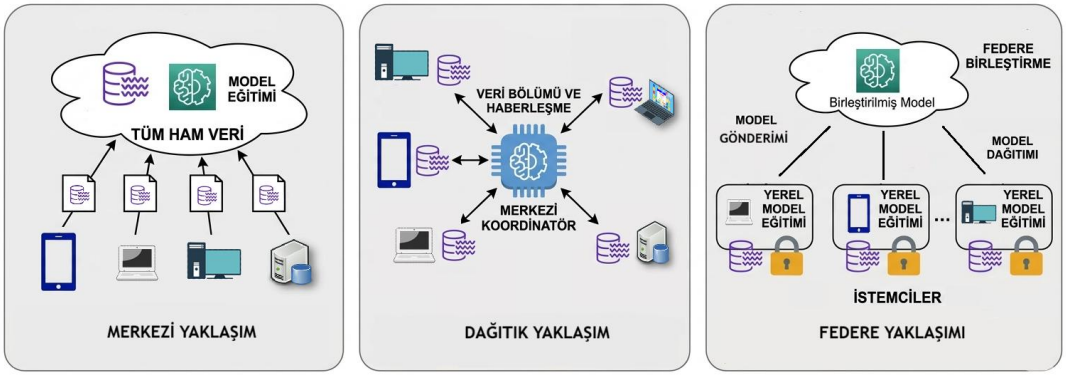
Geleneksel dağıtık öğrenme mimarileri donanımsal ve hesaplama açısından büyük avantajlar sağlamaktadır. Ancak veri mahremiyeti kısıtlarını ortadan kaldıramamaktadır. Klasik dağıtık öğrenmede hesaplama yükü farklı düğümlere paylaştırılır. Modelin eğitimi ise merkezi bir sunucu tarafından koordine edilir. Ancak bu mimari, verilerin ağ üzerinde aktarılmasını zorunlu kılar. Çoğu durumda sunucunun verilere doğrudan erişmesi gerekir. Sonuç olarak veriler kendi kaynağında izole edilemez. Bu yalıtım eksikliği, doğrudan veri gizliliği ihlallerine zemin hazırlamaktadır [4].

Kişisel verilerin izinsiz kullanımı küresel boyutta endişelere yol açmıştır. Bu durum hukuk ve teknoloji alanlarında köklü bir değişimi tetiklemiştir. Sonuç olarak çeşitli yasal düzenlemeler yürürlüğe girmiştir. Avrupa Birliği Genel Veri Koruma Tüzüğü (GDPR), Kaliforniya Tüketici Gizliliği Yasası (CCPA) ve Türkiye'deki Kişisel Verilerin Korunması Kanunu (KVKK) bu yasaların temel örnekleridir [5]. Bu düzenlemeler "Tasarım Yoluyla Gizlilik" (Privacy by Design) ilkesini zorunlu kılmıştır [6]. Özellikle GDPR'ın 17. maddesi "Unutulma Hakkı"nı (Right to be Forgotten) güvence altına almıştır [7]. Bu hak, bireylere verilerinin sistemlerden silinmesini talep etme yetkisi tanır. Böylece veri mahremiyeti isteğe bağlı bir özellik olmaktan çıkmış yasal bir zorunluluk haline gelmiştir.

Veri gizliliğini merkeze alan bu yasal zorunluluklar makine öğrenmesi alanında kendini göstermiştir. Bu doğrultuda literatüre Federe Öğrenme (Federated Learning - FL) mimarisi

kazandırılmıştır [8]. Federe öğrenme, gizlilik odaklı bir dağıtık öğrenme yaklaşımıdır. Bu mimaride kullanıcı verileri cihazdan/istemciden dışarı çıkarılmaz. Modeller doğrudan cihaz içinde, yerel olarak eğitilir. Ağ üzerinden yalnızca model güncellemeleri merkezi sunucuya aktarılır. Sunucu, istemcilerden gelen bu güncellemeleri birleştirerek yeni bir global model oluşturur. Oluşturulan bu güncel global model, tekrar tüm istemcilere dağıtılır. İstemciler yeni modeli teslim alır ve kendi yerel verileriyle eğitimi tekrarlar. İstemci ve sunucu arasındaki bu karşılıklı veri alışverişi iletişim turu olarak adlandırılır [9]. İletişim turları, global model hedeflenen doğruluk oranına veya yakınsama noktasına ulaşana kadar devam eder. Bu ardışık süreç sayesinde ham verinin gizliliği sağlanmış olur.

Makine öğrenmesinde kullanılan merkezi, dağıtık ve FL yaklaşımlarının yapısal ayrımları Şekil 1.1. içerisinde özetlenmiştir [10] .



Şekil 1.1. Merkezi, Dağıtık ve Federe öğrenme yaklaşımları arasındaki yapısal farklılıklar.

FL mimarisi eğitim aşamasında veri gizliliğini başarıyla sağlar. Ancak sistemin mevcut yapısı, yasal düzenlemelerin getirdiği Unutulma Hakkını doğrudan karşılayamaz. Sisteme katkı sağlayan bir istemci bu mimariden tamamen ayrılmak isteyebilir. Kullanıcılar kişisel verilerinin silinmesini yasal olarak talep edebilir. Ayrıca verilerin sistemdeki yasal saklama süresi dolmuş olabilir. Tüm bu senaryolarda, ilgili verinin global model üzerindeki ezberlenmiş etkilerinin tamamen ortadan kaldırılması zorunludur. Çünkü modern makine öğrenmesi algoritmaları eğitim verilerini ezberleme eğilimi gösterir. Kötü niyetli saldırganlar bu zafiyeti kullanarak modele yönelik güvenlik ihlalleri gerçekleştirir. Üyelik çıkarım saldırıları (Membership inference attacks) ve Model tersine mühendisliği (Model inversion attacks) bu ihlallerin başında gelir [11], [12]. Saldırganlar bu yöntemlerle sadece global model güncellemelerini inceleyerek orijinal eğitim verisini kısmen de olsa açığa çıkarabilir. Dolayısıyla, kullanıcının cihazındaki veriyi silmesi mahremiyet için tek başına yeterli değildir. O verinin modelin hafızasından da algoritmik olarak silinmesi gerekmektedir.

Merkezi veya dağıtık sistemlerde bir verinin etkisini modelden tamamen silmenin geleneksel bir yöntemi vardır. Bu yöntem, hedeflenen veri dışarıda bırakılarak modelin sıfırdan yeniden eğitilmesidir [13]. Teorik olarak bu yaklaşım kesin unutma sağlar. Ancak federe mimarilerinin kendine has dinamikleri bu geleneksel yöntemi pratik açıdan uygulanamaz kılar. Federe ağlarda iletişim maliyetleri oldukça yüksektir. İstemcilerin bant genişliği kısıtlıdır. Üstelik veriler cihazlar arasında bağımsız ve eş dağılımlı değildir [14]. Bu mimaride gelen her silme talebinde modeli baştan eğitmek uygulanabilir bir çözüm olamaz. Zaman, hesaplama gücü ve enerji tüketimi açısından devasa darboğazlar yaratır.

Sıfırdan eğitme zorunluluğunun yarattığı bu çözümsüzlük, literatürü yeni ve verimli bir mekanizma arayışına itmiştir. İşte bu sorunları aşmak ve silme taleplerini dinamik olarak yönetebilmek için Makine Unutması (Machine Unlearning - MU) kavramı çözüm olarak geliştirilmiştir [15]. MU, modeli baştan eğitme yerine, hedeflenen verinin model üzerindeki ezberlenmiş etkilerini algoritmik olarak silmeyi hedefler. Matematiksel işlemlerle geriye dönük bir unutma süreci tasarlar. Böylece büyük ölçekli modeller baştan eğitilmeden, sadece ilgili verinin izleri modelden verimli ve güvenli bir şekilde temizlenir.

Merkezi sistemlerde MU yönetilebilir bir süreçtir. Ancak FL mimarisinde bu işlemi gerçekleştirmek yapısal olarak çok daha karmaşıktır [16]. Çünkü federe mimaride merkezi sunucu istemcilerin verilerine erişemez. Federe mimarinin getirdiği bu yapısal zorluk, son yıllarda araştırmacıların ilgisini büyük ölçüde artırmıştır. Literatürde bu problemi çözmeye yönelik çok çeşitli federe mimaride unutma yöntemleri geliştirilmiştir [17].

Bu tez çalışması, FL ortamındaki unutma sürecinin getirdiği yapısal zorlukları bütüncül bir yaklaşımla ele almaktadır. Temel amaç veri gizliliğini, model başarımını ve Açıklanabilir Yapay Zeka (Explainable Artificial Intelligence - XAI) [18] temelli sistem şeffaflığını eş zamanlı olarak sağlayan yeni bir makine unutması mimarisi geliştirmektir. Bu doğrultuda, literatürdeki yaklaşımlara alternatif sunulmaktadır. Önerilen çözüm, kural tabanlı ilişkisel sınıflandırma algoritmaları üzerine inşa edilmiştir. Çalışmanın ilk aşamasında, literatüre özgün bir federe birleştirme algoritması olan Federated Classification Based on Associations (FedCBA**)** [1] kazandırılmıştır. Ardından, bu altyapı üzerinde veriden habersiz çalışabilen federe unutma yaklaşımı Federated Forgetting (FedFORGET) tasarlanmıştır.

Geliştirilen algoritmaların başarım ölçümleri ve doğrulama süreçleri aşamalı bir yaklaşımla gerçekleştirilmiştir. İlk olarak, önerilen kural birleştirme algoritması FedCBA'nın sınıflandırma performansı kapsamlı deneylerle analiz edilmiştir. Elde edilen başarılı bulguların ardından, ikinci aşamaya geçilmiş ve FedFORGET yaklaşımının unutma verimliliği ile model doğruluğu üzerindeki etkileri test edilmiştir. Geliştirilen mimarinin

başarım analizi yalnızca teorik simülasyonlarla sınırlandırılmamıştır. Sistem, Google Cloud tabanlı uygulama programlama arayüzü (Application Programming Interface − API) altyapısında ve Raspberry Pi uç cihazlarında uçtan uca test edilmiştir. Bu deneysel kurgular önerilen algoritmaların gerçek dünya senaryolarında uygulanabilirlik ve verimlilik sunduğunu göstermektedir.

1.1. Problem Tanımı

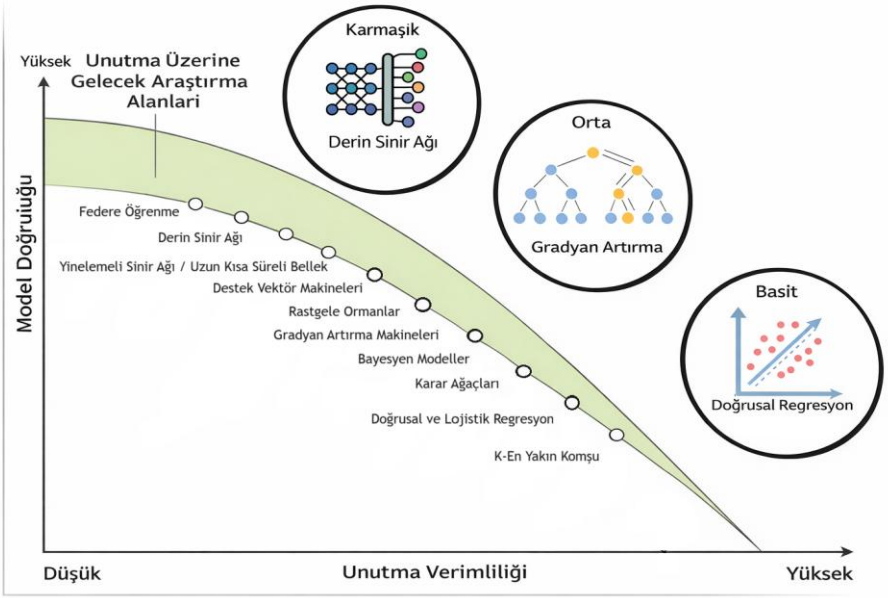
Federe unutma, geleneksel makine unutmasına kıyasla benzersiz yapısal zorluklar barındırır [16], [17]. Literatürde öne çıkan ve bu tez çalışmasının çözmeye odaklandığı temel problemler şu şekildedir:

- **Bilgi Yayılımı (Knowledge Permeation):** FL mimarisinde bir istemcinin verisine dair bilgi, global modelin güncellenmesi ve dağıtılması aracılığıyla diğer tüm istemcilere yayılmaktadır. Bu yayılım, modelin performansını ve tutarlılığını koruyarak spesifik bir veriyi silmeyi oldukça karmaşık hale getirmektedir.

- **Veri İzolasyonu Kısıtı (Data Isolation):** Federe sistemler, mahremiyet gereği veriyi istemci cihazında izole eder. Doğrudan veri erişiminin olmaması, veriye ihtiyaç duyan geleneksel MU algoritmalarının bu ortama uyarlanmasını engellemektedir.

- **Doğrulama (Verification) Eksikliği:** Unutma işleminin gerçekten yapılıp yapılmadığının doğrulanması kritik bir güvenlik problemidir. Literatürde sıklıkla kullanılan yapay sinir ağları gibi kara kutu modeller, standart ve şeffaf doğrulama yöntemlerinin geliştirilmesini kısıtlamaktadır.

- **Çoklu Ödünleşim (Trade-off) Dengesi:** Federe unutma algoritmaları geliştirilirken gizlilik, model doğruluğu ve hesaplama verimliliği arasında sürekli bir çatışma mevcuttur. Mevcut yöntemler, sıfırdan eğitme maliyetlerinden kaçınırken genellikle bu üç hedefi aynı anda sağlamakta başarısız olmaktadır. Literatürdeki makine öğrenmesi modellerinin karmaşıklığı ile unutma verimliliği arasındaki ters orantı problemi Şekil 1.2. grafiğinde açıkça görülmektedir [17]. Şekle göre modelin mimari karmaşıklığı arttıkça, doğruluk oranı yükselmekte ancak MU algoritmalarının hesaplama verimliliği azalmaktadır.



Şekil 1.2. Makine öğrenmesi modellerinde unutma verimliliği ile model doğruluğu arasındaki yapısal ödünleşim analizi [17].

1.2. Çalışmanın Arka Planı ve Çözüm Yaklaşımı

Kişisel verilerin gizliliğine yönelik yasal düzenlemeler makine öğrenmesi modellerinde "Unutulma Hakkı" kavramını yasal bir zorunluluk haline getirmiştir. Bu durum, eğitim verilerinin etkisini modellerden silmeyi amaçlayan FL mimarisinde MU yaklaşımlarının doğmasına yol açmıştır. Ancak literatürdeki mevcut algoritmalar, Bölüm 1.1'de belirtilen veri izolasyonu ve bilgi yayılımı problemlerini aşmakta zorlanmaktadır. Özellikle derin öğrenme gibi kara kutu modellerine dayalı yaklaşımlar, unutma sürecinin doğrulanabilmesi açısından şeffaflık sunamamaktadır.

Bu problemlerin aşılabilmesi için literatürde yeni bir mimari ihtiyacı doğmuştur. Bu tez çalışmasında, federe unutma sistemleri için mevcut gradyan ve ağırlık tabanlı kara kutu modellere alternatif olarak kural tabanlı ilişkisel sınıflandırma algoritmalarına dayalı bir çözüm yaklaşımı geliştirilmiştir. Önerilen mimari, karar mekanizmalarını açık kurallar dizisi olarak sunduğu için literatürde kendinden yorumlanabilir/açıklanabilir (intrinsically interpretable) [19] XAI modelleri kategorisinde yer almaktadır. Geliştirilen bu yaklaşım, harici bir açıklanabilirlik aracına ihtiyaç duymadan, doğrudan modelin kendi şeffaf iç yapısı sayesinde unutma işleminin izlenebilmesine ve doğrulanmasına imkan tanımaktadır.

1.3. Çalışmanın Amacı ve Motivasyonu

Bu tez çalışmasının temel amacı, literatüre yeni bir federe unutma mimarisi sunmaktır. Çalışmanın motivasyonunu, mevcut MU yaklaşımlarının federe ağlarında karşılaştığı yapısal kısıtlar oluşturmaktadır. Özellikle, MU için geliştirilmiş gradyan tabanlı kara kutu modelleri (yapay sinir ağları ve derin öğrenme mimarileri), unutma sürecinin doğrulanabilmesi için gereken şeffaflığı sağlayamamaktadır. Diğer yandan, hedeflenen veriyi modelden silmek amacıyla uygulanan sıfırdan eğitme işlemleri, federe ağlarda yüksek iletişim ve hesaplama maliyetlerine yol açmaktadır.

Bu kısıtlar doğrultusunda, literatürdeki kara kutu yaklaşımlarına alternatif olarak ilişkisel sınıflandırma algoritmalarını [20] temel alan bir çözüm hedeflenmiştir. Çalışmanın odak noktası, orijinal veriye erişim gerektirmeyen, model başarımını koruyan ve karar mekanizmalarını kurallar aracılığıyla şeffaf bir şekilde açıklayabilen bir federe unutma mekanizması geliştirmektir.

1.4. Literatüre Temel Katkılar

Bu tez çalışmasının FL ve MU literatürüne sunduğu başlıca özgün katkılar şunlardır:

- **Literatür Analizi ve Sistematik Haritalama:** MU ve FL alanlarındaki mevcut çalışmalar analiz edilmiştir. MU ve FL kavramları ile bu alanlardaki güncel teknikleri ayrı ayrı inceleyen kapsamlı derleme makaleleri sunulmuştur. Bununla birlikte, federe mimarilerde makine unutması süreçleri üzerine sistematik bir haritalama çalışması yürütülmüştür. Çalışma sonucunda alandaki mevcut durum, uygulama alanları ve araştırma boşlukları belirlenmiştir.

- **Federe Birleştirme Algoritması - FedCBA:** Önceki çalışmalarda literatüre Data Unaware Classification Based on Associations - du-CBA [21] adıyla sunulan ilişkisel sınıflandırma tabanlı kural birleştirme algoritması, bu tez çalışmasında performans optimizasyonları yapılarak ileri bir seviyeye taşınmıştır. Algoritma, alan literatüründeki yaygın isimlendirme standartlarına uygun olarak FedCBA [1] adıyla güncellenmiştir. Model parametreleri yerine kuralları birleştiren bu alternatif altyapı, uluslararası görünürlüğünü ve erişilebilirliğini artırmak amacıyla İngilizce dilinde yayımlanarak küresel literatüre kazandırılmıştır [1].

- **Makine Unutması Yöntemi - FedFORGET:** Geliştirilen kural tabanlı birleştirme altyapısı üzerine, veriye ihtiyaç duymayan ve özgün bir federe unutma yaklaşımı olan FedFORGET literatüre kazandırılmıştır. Bu yöntemle, FL ortamındaki makine unutması problemi ilişkisel sınıflandırma temelli bir mimariyle çözülmüştür.

1.5. Tezin Organizasyonu

Bu tez çalışması toplam beş ana bölümden oluşmaktadır. Birinci bölümde problem tanımı, çalışmanın arka planı, çözüm yaklaşımı ve literatüre katkılar sunulmuştur. İkinci bölümde makine unutması, federe öğrenme, federe unutma, ilişkisel sınıflandırma ve XAI konularındaki teorik altyapı ile bu konular ile ilgili literatür çalışmaları incelenmiştir. Üçüncü bölümde, önerilen FedCBA kural birleştirme algoritması ve FedFORGET makine unutması mimarisinin tasarımı detaylandırılmıştır. Dördüncü bölümde önerilen yöntemlerin deneysel başarım sonuçları analiz edilmiştir. Beşinci bölümde ise elde edilen bulgular tartışılmış ve gelecek çalışmalar için öneriler sunulmuştur.

2. TEORİK ALTYAPI VE İLGİLİ ÇALIŞMALAR

Bu bölümde, tez kapsamında önerilen mimarinin üzerine inşa edildiği temel kavramlar anlatılmakta ve bu alanlarda literatürde yer alan ilgili çalışmalar incelenmektedir. Çalışmanın kavramsal çerçevesini oluşturan alt başlıkların organizasyonu şu şekildedir: Bölüm 2.1 ve Bölüm 2.2'de, sırasıyla MU ve FL kavramlarının teorik altyapıları sunulmaktadır. Bu bölümlerde temel tanımların yanı sıra, son beş yıla ait araştırma ve derleme makaleleri üzerinden güncel çalışma alanları kıyaslama tablolarıyla verilmektedir. Bölüm 2.3'te, bu iki alanın entegrasyonundan doğan federe unutma mimarisi incelenmektedir. Merkezi sunucu ile istemciler arasındaki unutma süreçleri literatürdeki güncel yaklaşımlar eşliğinde değerlendirilmektedir. Bölüm 2.4'te, bu tez çalışmasının algoritmik temelini oluşturan ilişkisel sınıflandırma yöntemleri tanıtılarak çeşitli veri setleri üzerindeki performans analizleri sunulmaktadır. Son olarak Bölüm 2.5'te, unutma sürecinin doğrulanabilmesi problemi ekseninde açıklanabilir yapay zeka kavramı ele alınmaktadır.

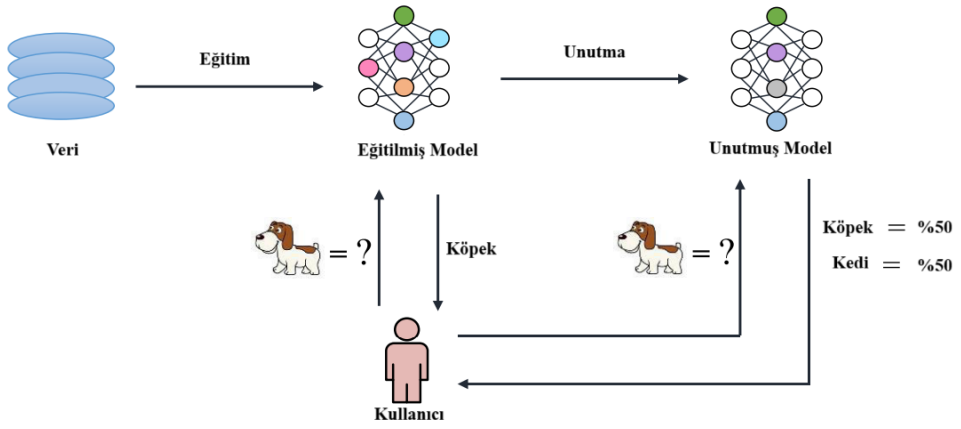
2.1. Makine Unutması (Machine Unlearning)

Günümüzde veri merkezli teknolojiler büyük bir hızla gelişmektedir. Bu bağlamda, makine öğrenmesi algoritmalarıyla eğitilen modeller, karmaşık karar alma süreçlerini başarıyla otomatikleştirerek birçok bilimsel alanda ve sektörde köklü değişimlerin önünü açmıştır. Sağlıktan finans sektörüne kadar geniş bir yelpazede, bu modeller tahmine dayalı analitik ve stratejik karar alma mekanizmalarının merkezine yerleşmiştir. Ancak modellerin yüksek başarıma ulaşmak için büyük hacimli kullanıcı verileriyle eğitilmesi mahremiyet, etik ve veri yönetimi konularında endişeleri beraberinde getirmiştir. Bu endişelere yanıt olarak yürürlüğe giren GDPR ve KVKK gibi yasal düzenlemeler, bireylere "unutulma hakkı" tanıyarak bağlayıcı yükümlülükler getirmiştir. Ancak, mahremiyetin korunmasına yönelik bu yasal düzenlemelerin sadece hukuki zeminde kalması teknik açıdan yeterli bir çözüm sunmamaktadır. Veri tabanı sistemlerinde bir kullanıcıya ait verilerin fiziksel olarak imha edilmesi yasal yükümlülükleri karşılamak için yeterli kabul edilse dahi makine öğrenmesi modellerinin doğası bu yaklaşımı geçersiz kılmaktadır. Zira makine öğrenmesinde kullanılan algoritmalar, eğitim süreçleri boyunca kullandıkları verilerin istatistiksel örüntülerini ve karakteristik özelliklerini kendi iç yapılarına kodlamaktadır. Dolayısıyla, veri sistemden tamamen silinse dahi, önceden eğitilmiş bir modelin çıktıları üzerinden bu hassas bilgilere tekrar çıkarım yapılabilme riski devam etmektedir. Tam da bu nedenle, kişisel verilerin yalnızca depolama birimlerinden fiziksel olarak silinmesi veya erişime kapatılması günümüzde yeterli görülmemektedir. Veri mahremiyetinin tam olarak

sağlanabilmesi için, yapay zeka modelinin de hedeflenen veriye dair edindiği tüm bilgiyi ve etkiyi kalıcı olarak unutması talep edilmektedir [22].

Ancak bu model unutması sürecinin algoritmik olarak hayata geçirilmesi, özellikle büyük ölçekli derin öğrenme mimarilerinde karmaşık ve hesaplama maliyeti yüksek bir problemdir [23]. Bu konuda literatürdeki yaklaşım, silinmesi hedeflenen verinin eğitim setinden tamamen çıkarılarak modelin en baştan yeniden eğitilmesidir (Retrain). Fakat bu yaklaşım, yüksek zaman maliyeti ve yoğun donanım kaynağı tüketimi nedeniyle günümüzün büyük veri odaklı dinamik sistemlerinde sürdürülebilir ve uygulanabilir bir çözüm değildir [24]. Bu problemlere çözüm olarak makine unutması kavramı ortaya çıkmıştır. Bu yaklaşım, hedeflenen verilerin eğitilmiş model üzerindeki etkisini tamamen silmeyi ve modelin bu verileri hiç tecrübe etmemiş gibi davranmasını amaçlamaktadır [25].

Şekil 2.1, MU sürecinin temel adımlarını göstermektedir. Önce eğitilmiş hedef modele unutma talebi gelmektedir. Daha sonra unutma algoritmaları çalıştırılır. Sonuç olarak, silinmek istenen verinin etkisinden arındırılmış unutmuş model oluşturulur. Sürecin pratik işleyişinde unutma işleminin başarısı, hedef bilginin silinmesi ile modelin genel performansının korunması arasındaki dengeyle ölçülür. Teorik olarak bir kullanıcı sistemi test ettiğinde öncelikle orijinal eğitilmiş modele silinmesi istenen veriye dair bir sorgu yöneltir ve modelden beklenen yanıtı alır. Ancak aynı sorgu, unutma işlemi tamamlandıktan sonra elde edilen unutmuş modele sorulduğunda, modelin artık bu veriyi hatırlamaması ve önceki cevabı üretmemesi beklenir. Buradaki asıl hedef, modelin genel tahmin başarımından ve doğruluğundan hiçbir ödün vermeden, yalnızca kullanıcının silinmesini talep ettiği veriyi başarılı bir şekilde unutmasıdır. Bu temel mimari akıllı şehirler, sağlık IoT uygulamaları, uç bilişim ve büyük dil modelleri gibi veri hassasiyeti yüksek alanlarda kullanılmaya başlamıştır.



Şekil 2.1. Makine unutması sürecinin temel işleyişi.

MU kavramının teorik altyapısı ve temel işleyiş mimarisi incelendiğinde, sürecin sadece veri silme işlemi olmadığı aksine mahremiyet ile model performansı arasında hassas bir denge gerektirdiği anlaşılmaktadır. Bu bağlamda Tablo 2.1. ile genel hatlarıyla özetlenen MU sürecinin sunduğu temel avantajlar ve dezavantajlar detaylandırılmıştır.

Tablo 2.1. Makine unutması sürecinin avantaj ve dezavantajları.

Avantajlar	Dezavantajlar
Kullanıcı Gizliliği ve Yasal Uyum	Hesaplama Maliyeti
Model Güvenliğinin Artırılması	Performans Kaybı Riski
Modelin Güncel Tutulması	Model Mimarisine Bağımlılık
Hatalı Verilerin Düzeltilmesi	Veri Bağımlılıkları ve Zincirleme Etkiler

Veri koruma yasaları ile unutulma hakkı güvence altına alınmıştır. MU yöntemleri ile bu hak sadece hukuki zorunluluk olmaktan çıkarılıp teknik bağlamda işlevsel hale getirilmiştir. Silinmesi talep edilen verinin model üzerindeki etkisi ortadan kaldırılarak kullanıcı gizliliği algoritmik seviyede garanti altına alınmaktadır.

MU yöntemleri, eğitim aşamasında kasıtlı eklenen kötü niyetli veriler veya zafiyetler tespit edildiğinde, modeli sıfırdan eğitmek yerine yalnızca bu zehirli verilerin etkisini silme imkanı sunmaktadır. Bunun yanı sıra MU, zamanla geçerliliğini yitiren, dağılımı değişen veya baştan hatalı etiketlenmiş verilerin model parametrelerinden ayıklanmasına imkan tanıyarak sistemin her daim dinamik, doğru ve güncel kalmasını sağlamaktadır.

MU yönteminin sunduğu bu avantajların yanı sıra dikkate alınması gereken bazı dezavantajları ve sınırlamaları da mevcuttur. Özellikle milyarlarca parametreye sahip karmaşık derin öğrenme mimarilerinde, spesifik bir verinin istatistiksel etkisini geri almak oldukça zorlu bir optimizasyon problemidir. Öyle ki bazı durumlarda unutma algoritmalarının çalıştırılması, modeli sıfırdan yeniden eğitmek kadar yüksek donanımsal ve zamansal maliyetler doğurabilmektedir. Yani unutma sürecinin fizibilitesi doğrudan kullanılan makine öğrenmesi algoritmasının yapısına bağlıdır. Örneğin, K-En Yakın Komşu algoritması gibi belleğe dayalı modellerde bir veriyi unutmak sadece onu veri tabanından silmek kadar basitken derin sinir ağlarında bir verinin etkisi milyonlarca ağırlığa dağıldığı için bu etkiyi izole edip temizlemek son derece karmaşıktır.

Veri unutma işlemi, modelin hedeflenmeyen ancak faydalı olan diğer bilgileri de kaybetmesine ve bunun sonucu olarak genel tahmin doğruluğunun düşmesine yol açabilmektedir. Ancak buradaki temel tasarım felsefesi performanstan neden ödün verildiği üzerinedir. Veri gizliliğini sağlamak ve bağlayıcı yasal düzenlemelere uymak, modelin genel doğruluğundaki bu düşüşü göze almayı zorunlu kılmaktadır. Göze alınan bu

performans kaybının teknik altyapısına bakıldığında ise, eğitim kümesindeki bazı verilerin modelin karar sınırlarını belirlemede diğerlerinden çok daha baskın olduğu görülmektedir. Bu tür kritik verilerin sistemden çıkarılması, zincirleme bir etki yaratarak modelin tamamen alakasız gibi görünen başka sınıflardaki genelleme yeteneğini de beklenmedik biçimde bozabilmektedir.

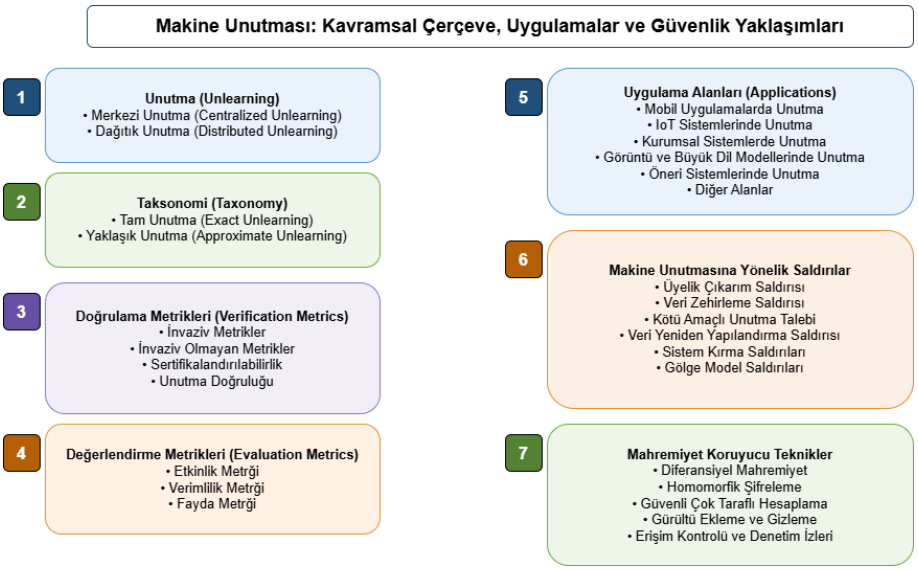
İncelenen bu avantaj ve dezavantajlar doğrultusunda, son yıllarda MU sürecini hem donanımsal hem de algoritmik açıdan daha verimli hale getirmek amacıyla çeşitli yaklaşımlar önerilmiştir [26]. Literatürde bu teknikler temel olarak; Tam Unutma (Exact Unlearning), Yaklaşık Unutma (Approximate Unlearning), Sertifikalı Unutma (Certified Unlearning) ve Doğrulanabilir Unutma (Verifiable Unlearning) başlıkları altında incelenmektedir [24], [25]. Her bir yöntem; doğruluk, hız ve yasal uyumluluk bağlamında farklı avantajlar ve kısıtlamalar barındırmaktadır. Bu teknikler yalnızca yasal veri silme taleplerini karşılamakla kalmamaktadır. Öte yandan model yanlılıklarını gidermek, model zehirlenmesi saldırılarına karşı direnç sağlamak ve yapay zeka etiğini desteklemek amacıyla da etkin bir şekilde kullanılmaktadır [27], [28], [29].

Güncel araştırmalar, MU tekniklerinin kaynak kısıtlı ve dağınık cihazların rol aldığı IoT ekosistemlerine entegrasyonuna da yoğunlaşmaktadır. Uç ve gömülü cihazların heterojen yapısı, geniş ölçeği ve sınırlı işlem kapasitesi veri gizliliği zorluklarını daha da derinleştirmektedir. Bu nedenle uç bilişim tabanlı uygulamalar, hesaplama açısından hafif ve iletişim bakımından verimli yeni nesil MU çözümlerine ihtiyaç duymaktadır. Örneğin, giyilebilir sağlık sensörlerinde veya güvenlik sistemlerinde kullanıcıların onaylarını anında geri çekebilmesi, sistemin genel model başarımından ödün vermeden ilgili veriyi hızla silebilmesini zorunlu kılmaktadır.

FL mimarilerinde verinin yerel cihazlarda tutulması, MU problemini çok daha karmaşık bir boyuta taşımaktadır [30]. Bu durum, FL modelleri içerisinde mahremiyeti koruyan unutma mekanizmalarının etkinliğini ve verimliliğini artırmaya yönelik araştırmalara hız kazandırmıştır. Buna ek olarak, hedeflenen verinin modelden gerçekten silinip silinmediğinin saptanması da bağımsız bir araştırma alanı haline gelmiştir. Bu kapsamda, modelin unutma işlemini başarıyla gerçekleştirip gerçekleştirmediğini matematiksel olarak analiz etmek için unutma doğrulaması ve etki fonksiyonu gibi yöntemler kullanılmaktadır [27], [28].

Tüm bu kavramsal çerçeve ve güncel literatür ışığında; MU yaklaşımlarının temel algoritmaları, değerlendirme metrikleri ve uygulama alanlarını kapsayan kapsamlı

taksonomi Şekil 2.2. ile şematize edilmiş olup, ilgili alt bileşenler detaylı olarak analiz edilmiştir.



Şekil 2.2. Makine Unutması: Teknikleri, Metrikleri ve Uygulama Alanları.

Şekil 2.2. ile şematize edilen MU taksonomisinde yer alan tüm alt bileşenler, uygulama alanları ve teknik kavramlar aşağıda kapsamlı bir biçimde ele alınmaktadır.

Merkezi Unutma (Centralized Unlearning): Merkezi Unutma, unutulması istenen verilerin sistemden çıkarılmasının ardından, modelin kalan tüm eğitim verileriyle sıfırdan yeniden eğitilmesi esasına dayanır. Bu yöntem teorik olarak tam bir unutma garantisi sunar ve özellikle GDPR gibi veri koruma düzenlemeleri bağlamında oldukça güçlüdür. Ancak bu yaklaşım, büyük veri setleri veya derin sinir ağı modelleri ile çalışırken ciddi hesaplama maliyetlerine yol açmaktadır. Buna ek olarak, orijinal eğitim verilerine sürekli erişim gerektirdiği için pratik uygulamalarda kısıtlayıcı bir rol oynamaktadır [31].

Dağıtık Unutma (Distributed Unlearning): FL, kullanıcı verilerinin yerel cihazlarda tutulduğu ve merkezi bir sunucuya yalnızca model güncellemelerinin iletildiği dağıtık bir öğrenme yaklaşımıdır. FL, veri mahremiyeti açısından avantajlar sunsa dahi, unutma süreçlerini önemli ölçüde karmaşıklaştırmaktadır. Bir FL ortamında, model güncellemeleri heterojen ve asenkron istemcilerden gelir. Bu durum spesifik bir kullanıcının katkısını modelden çıkarmayı teknik açıdan zorlaştırır [32]. Bölüm 2.3. Federe Öğrenme Ortamında Makine Unutması (Federated Unlearning) başlığı altında bu konu daha detaylı anlatılmaktadır.

Tam Unutma (Exact Unlearning): MU alanında tam unutma, silinmesi hedeflenen

verilerin modelin parametrelerinden ve hafızasından kalıcı ve kesin bir biçimde çıkarılmasını ifade eder. Tüm modeli sıfırdan eğitmeye gerek kalmadan bu işlemi verimli bir şekilde gerçekleştirebilmek için literatürde önerilen en temel yaklaşımlardan biri SISA (Bölümlenmiş, İzole Edilmiş, Dilimlenmiş, Birleştirilmiş - Sharded, Isolated, Sliced, Aggregated) mimarisidir. SISA yöntemi, veri setini daha küçük parçalara böler, her bir parçayı ayrı modeller üzerinde eğitir ve nihai sonuçları üretmek için bu modelleri birleştirir [25], [33]. Bu yaklaşımda, silinmesi gereken veri yalnızca ait olduğu parçada unutulur. Böylece tüm sistemin yeniden eğitilmesine gerek kalmadan unutma işleminin gerçekleştirilmesi mümkün hale gelir. Fakat bu parçalı yapıdan dolayı eğitim süreci daha karmaşık bir hal alabilir ve dengesiz veri dağılımı model performansını olumsuz yönde etkileyebilir.

Yaklaşık Unutma (Approximate Unlearning): Yaklaşık unutma yöntemleri, tam unutma garantisi sunmaksızın, unutma işlemini çok daha düşük bir hesaplama maliyetiyle gerçekleştirmeyi amaçlar. Bu yaklaşımlar veri odaklı ve model odaklı yöntemler olmak üzere iki ana kategoriye ayrılmaktadır.

Veri odaklı yöntemler, bir veri noktasının model üzerindeki etkisini etki fonksiyonu gibi istatistiksel analizler kullanarak tahmin eder ve bu etkiyi tersine mühendislikle ortadan kaldırır [30]. Model odaklı yaklaşımlar ise bilgi damıtma veya gradyan tersine çevirme gibi teknikler kullanarak modelin doğrudan yapısına müdahale eder ve modeli yeniden yapılandırır. Bu yöntemler zaman ve donanım kaynaklarından tasarruf sağlasa dahi, unutma işleminin eksiksiz gerçekleşip gerçekleşmediğini doğrulamak oldukça zordur. Tablo 2.2. içerisinde bahsedilen iki yaklaşımın kıyaslaması yer almaktadır.

Tablo 2.2. Kesin unutma ve yaklaşık unutma yöntemlerinin karşılaştırması [34].

Kategori	Avantajlar	Dezavantajlar
Tam Unutma	Daha etkili bir unutma süreci sunar. Gizlilik sızıntısı riskini tamamen ortadan kaldırır.	Çok yüksek zaman ve hesaplama maliyeti gerektirir. Karmaşık modellerde uygulanması zordur. Algoritmik tasarımı teorik olarak zorlayıcıdır.
Yaklaşık Unutma	Derin öğrenme gibi daha karmaşık modellere kolayca uygulanabilir. Kesin unutmaya kıyasla minimum zaman ve işlem maliyeti gerektirir. Pratik olarak başarılması çok daha kolaydır.	Modelin parametreleri içerisinde silinen veriye ait hassas bilgi ve izlerin kalma riski bulunur. Yani tam güvenlik sağlamaz.

Doğrulama Metrikleri (Verification Metrics): Unutma uygulamalarının güvenilirliği,

yalnızca işlemin teknik olarak uygulanmasıyla değil aynı zamanda silinmesi hedeflenen verinin modelden gerçekten arındırılıp arındırılmadığının test edilmesiyle de ölçülmektedir. Bu doğrulama sürecinde kullanılan metrikler, invaziv (müdahaleci) ve invaziv olmayan olmak üzere iki ana kategoriye ayrılmaktadır.

İnvaziv Metrikler (Invasive Metrics): İnvaziv metrikler, modelin iç yapısına (parametreler, aktivasyon fonksiyonları veya gradyanlar) doğrudan erişim sağlayarak unutma işlemini analiz eder. Silinen veriyle ilişkili ağırlıklardaki değişimler doğrudan gözlemlenerek unutmanın derecesi saptanabilir. Teorik olarak güçlü bir doğrulama sunsalar da, model ağırlıklarına tam erişim gerektirmeleri pratik uygulanabilirliklerini büyük ölçüde sınırlar [34]. Bu yaklaşıma literatürdeki en belirgin örneklerden biri Özellik Enjeksiyonu Testi'dir (Feature Injection Test) [32]. Bu yöntemde, unutulması hedeflenen veriye ayırt edici sahte bir özellik eklenir ve unutma işlemi sonrasında bu özelliğe ait model ağırlığının sıfırlanıp sıfırlanmadığı doğrudan iç yapıda kontrol edilir.

İnvaziv Olmayan Metrikler (Non-invasive Metrics): İnvaziv olmayan metrikler, modelin mimarisinden bağımsız olarak yalnızca girdi ve çıktılara odaklanan kara kutu değerlendirme yöntemleridir. Unutulması istenen veri modele tekrar sunulduğunda tahmin güveninde yaşanan düşüş veya Üyelik Çıkarım Saldırılarına (Membership Inference Attacks - MIA) karşı gösterilen direnç, unutma işleminin başarısını ölçmek için yaygın olarak kullanılmaktadır [34]. Bu temel yöntemlere ek olarak doğrulama sürecinde Arka Kapı Saldırıları (Backdoor Attacks) ve Bilgi Sızıntısı (Information Leakage) analizleri de kullanılmaktadır [32]. Örneğin, eğitim sürecinde modele enjekte edilen tetikleyici verilerin unutma işleminden sonra etkisini yitirmesi veya Sınıflar Arası Karışıklık Testi (Interclass Confusion Test) ile çıktı uzayındaki karmaşıklık matrisinden bir unutma skoru hesaplanması, dışarıdan gözlemlenebilen stratejilerdir [32].

Sertifikalandırılabilirlik (Certifiability): Sertifikalandırılabilirlik, MU işleminin matematiksel bir kanıta dayandırılmasıdır. Yüksek maliyetli sıfırdan eğitime verimli bir alternatif sunan bu standart üretilen unutmuş modelin, hedef veri olmadan eğitilmiş modele ne kadar yakınsadığını teorik olarak sınırlandırır ve ispatlar [35]. SNARK veri yapıları ve Merkle ağaçları gibi mimariler kullanılarak, silinen bir öğenin modelden çıkarıldığı ve kalan verilerde bulunmadığı kriptografik düzeyde doğrulanabilir [32].

Unutma Doğruluğu (Forgetting Accuracy): Unutma doğruluğu, modelin kasıtlı olarak unuttuğu veriler üzerindeki tahmin performansını değerlendirir. Başarılı bir MU sürecinde, modelin silinen verilere ait doğruluk oranının rastgele tahmin seviyesine inmesi beklenmektedir. Geri kalan veri seti üzerindeki genel model performansın ise korunması

temel kriterdir [33]. Bu doğruluğu zamansal ve oransal olarak ifade edebilmek için Unutma Ölçümü (Forgetting Measuring) yaklaşımları kullanılır. Bu sayede bir verinin modelden başarıyla silindiğini kabul etmek için saldırı başarı eşiği referans alınır [32].

Değerlendirme Metrikleri (Evaluation Metrics): MU yaklaşımlarının değerlendirilmesinde yalnızca verinin sistemden arındırılması şartı yeterli bir kıstas değildir. Sürecin etkinliğini, kaynak tüketimi açısından verimliliğini ve geriye kalan veriler üzerindeki model faydasını eş zamanlı olarak ölçecek çok boyutlu bir yaklaşıma ihtiyaç duyulmaktadır. Bu doğrultuda etkinlik, verimlilik ve fayda metrikleri değerlendirme aşamasının temel taşlarını teşkil etmektedir [34].

Etkinlik Metriği (Effectiveness Metric): Etkinlik metriğinin temel amacı, modelin hedeflenen veriden tamamen arınıp arınmadığını doğrulamaktır. Unutma işlemi başarılıysa, model bu veriyi daha önce hiç tecrübe etmemiş gibi çalışır. Bu durumu doğrulamak için modelin çıktılarına bakılır. Beklenen sonuç modelin silinen veri üzerindeki doğruluğunun düşmesi ve artık bu veriye dair kararlı tahminler üretememesidir. Bu sayede yöntemin başarısı ölçülmüş olur [34].

Verimlilik Metriği (Efficiency Metric): Verimlilik metriklerinin temel amacı, unutma işleminin donanımsal ve zamansal maliyetlerini ölçmektir. "Unutulma hakkı" gereği veri silme işlemlerinin belirli bir süre zarfında tamamlanması yasal bir zorunluluktur. Dolayısıyla bu sürecin olabildiğince hızlı gerçekleşmesi, kullanıcılara sunulan hizmetin kesintisiz devam edebilmesi açısından önemlidir. Bu bağlamdaki performans analizleri; işlem süresi, CPU/GPU iş yükü ve bellek kapasitesi gibi parametreler dikkate alınarak yapılır. Literatürde, yaklaşık unutma yaklaşımlarının bu metriklerde yüksek verimlilik sağladığı, merkezi unutma yöntemlerinin ise ciddi donanımsal darboğazlar yaratarak düşük verimlilik sunduğu kanıtlanmıştır [34].

Fayda Metriği (Utility Metric): Fayda metrikleri, unutma işleminin ardından modelin silinmeyen yani geriye kalan veriler üzerindeki genel performansının ne ölçüde korunduğunu test eder. Modelden veri çıkarmak tahmin performansını düşürebileceğinden, unutmuş modelin gerçek dünya senaryolarında hala kullanılabilir olduğunun kanıtlanması şarttır. Bu değerlendirme sürecinde; Doğruluk (Accuracy), F1-Skor, Kesinlik (Precision) ve Duyarlılık (Recall) gibi temel metrikler hesaplanır. Etkili bir unutma yöntemi, hedef veriyi sistemden başarıyla silerken, model üzerindeki tahmin yeteneğinden ve genel faydasından ödün vermemelidir [34].

Literatürde geliştirilen MU yöntemlerini sistematik biçimde sınıflandırmak ve

karşılaştırabilmek için, problem tarafımızca iki temel eksende sınıflandırılmaktadır. Bunlar; sistemin hangi hedef veriyi unutması gerektiğini belirten Unutma Talepleri (Unlearning Requests) ve bu işlemin hangi performans veya güvenilirlik kısıtları altında gerçekleştirildiğini ortaya koyan Tasarım Gereksinimleridir (Design Requirements).

MU literatüründe, kullanıcılardan veya sistem gereksinimlerinden doğan unutma talepleri temelde şu şekilde kategorize edilmektedir [32]:

Öğe Çıkarımı (Item Removal): Eğitim verilerinden belirli örneklerin silinmesi işlemidir. MU alanındaki en yaygın talep türüdür.

Özellik Çıkarımı (Feature Removal): Gizlilik ihlalleri her zaman tek bir veri örneğinden kaynaklanmaz. Bazı durumlarda benzer özelliklere veya etiketlere sahip veri gruplarının bütünüyle silinmesi gerekebilir. Bu verileri geleneksel yöntemlerle tek tek sırayla silmek, hesaplama açısından yüksek maliyetler yaratır ve modelin genel performansını doğal olarak düşürür. Bu zorluğu aşmak amacıyla literatürde, tekil işlemler yerine optimize edilmiş toplu güncellemeler sağlayan matematiksel etki fonksiyonları kullanılmaktadır.

Sınıf Çıkarımı (Class Removal): Eğitilmiş bir makine öğrenmesi modelinden spesifik bir sınıfa ait tüm veri örneklerinin ve bu sınıfa dair öğrenilmiş temsillerin bütünüyle arındırılması sürecini tanımlamaktadır. Makine öğrenmesi algoritmalarının temel mimarisi çoğunlukla sınıflar arası ayırt edici özelliklerin ve karar sınırlarının optimizasyonuna dayandığından, bütünleşik bir sınıfın modelden çıkarılması işlemi, tekil veri örneklerinin silinmesine kıyasla teorik ve operasyonel açıdan çok daha yüksek bir karmaşıklık teşkil etmektedir.

Görev Çıkarımı (Task Removal): Yaşam boyu öğrenme modellerinde, modelin önceden öğrendiği spesifik bir görevi bellekten silmesi işlemidir. Öğrenilen görevlerin ardışıklığı model mimarisini doğrudan etkilediğinden, bir görevin silinmesi diğer görevlerin performansını bozarak yıkıcı unutma sorununa yol açabilir. Bu durumu önlemek adına, modelin eğitim sürecinde kalıcı ve geçici görevlerin sınırlarının baştan net bir şekilde belirlenmesi esastır.

Akış Çıkarımı (Stream Removal): Sisteme ardışık ve sürekli olarak iletilen veri silme taleplerinin yönetilmesi sürecidir. Genellikle çoklu kullanıcı sistemlerinde kullanıcıların ardışık olarak ayrılması veya veri setindeki zehirli örneklerin kademeli şekilde temizlenmesi amacıyla uygulanır. Bu silme talepleri önceden belirlenmiş sabit bir düzende

gerçekleştirilebileceği gibi, modelin anlık durumuna ve performansına bağlı olarak dinamik bir şekilde de işlenebilir.

Bir MU algoritmasının başarısı yalnızca veriyi modelden silme yeteneğiyle sağlanmaz. Ayrıca bu işlemin ne kadar verimli, güvenilir ve doğrulanabilir olduğunu belirleyen tasarım gereksinimleri ile ölçülmektedir. Bu temel kriterler literatürde şu şekilde sıralanmaktadır [32]:

Eksiksizlik / Tutarlılık (Completeness / Consistency): Başarılı bir unutma algoritması eksiksiz olmalıdır. Unutmuş model ile sıfırdan eğitilmiş model, tüm verilerde aynı tahmin sonuçlarını vermelidir. Bu tutarlılık, iki modelin çıktı uzayları arasındaki farkın minimize edilmesiyle sağlanabilmektedir.

Zamanındalık / Hız (Timeliness): Modeli sıfırdan eğitmek kusursuz bir çözüm olsa da çok zaman alır. Bu nedenle eksiksizlik ile işlem hızı arasında bir denge kurulmalıdır. Az miktarda veri silineceğinde hızlı unutma algoritmaları idealdir. Ancak çok fazla veri silinecekse, model performansının çökmesini engellemek için sıfırdan eğitim tercih edilmelidir.

Doğruluk (Accuracy): Unutma işlemi sonrasında model, test örneklerini doğru tahmin etmeye devam etmelidir. Modelin doğruluğu, sıfırdan eğitilmiş modelin veya orijinal modelin performansıyla kıyaslanabilir düzeyde kalmalıdır.

Hafiflik (Light-Weight): Unutma süreci hazırlıkları kapsamında model ağırlıkları ve geçici veriler saklanabilir. İyi bir algoritma düşük depolama alanı gerektirmeli, büyük veri kümeleriyle ölçeklenebilmeli ve ek hesaplama yüklerini en aza indirmelidir.

Kanıtlanabilir Garantiler (Provable Guarantees): Sıfırdan eğitim dışındaki unutma süreçleri doğası gereği yaklaşık sonuçlar verebilir. Bu nedenle, unutmuş modelin sıfırdan eğitilmiş modele olan yakınlığı matematiksel olarak sınırlandırılmalı ve teorik garantiler sunulmalıdır.

Modelden Bağımsızlık (Model-Agnostic): Geliştirilen unutma çerçevesi, tek bir yapıya bağlı kalmadan farklı makine öğrenmesi modellerine ve algoritmalarına uyarlanabilir olmalıdır.

Literatürdeki MU yaklaşımlarını özetleyen Tablo 2.3. aşağıda sunulmaktadır. Bu tabloda mevcut yöntemler; modelden bağımsız, modele özgü ve veri odaklı olmak üzere üç temel kategoride sınıflandırılmıştır. Tabloda her bir yöntemin tasarım gereksinimleri ve unutma

taleplerini ne ölçüde karşılayabildiği karşılaştırmalı olarak anlatılmaktadır. Bu doğrultuda diferansiyel gizlilik ve sertifikalı kaldırma gibi altyapıdan bağımsız çalışabilen teknikler modelden bağımsız yöntemler altında incelenmektedir. Yapay sinir ağları, ağaç tabanlı veya Bayes tabanlı modeller gibi belirli mimariler için tasarlanmış mekanizmalar modele özgü yöntemler olarak sınıflandırılmaktadır. Eğitim verisinin manipülasyonuna (veri bölmeleme, veri artırma vb.) dayanan yaklaşımlar ise veri odaklı yöntemler çatısı altında değerlendirilmektedir. Tabloda kullanılan sembollerden "✓" işareti ilgili yöntemin gereksinimi tam olarak desteklediğini, "✗" işareti desteklemediğini, "-" işareti ise kısmen veya dolaylı olarak desteklediğini göstermektedir.

Tablo 2.3. Literatüre kazandırılmış unutma yöntemlerinin karşılaştırılması.

Unutma Yöntemleri	**Unutma Tasarımı**		**Tasarım Gereksinimleri**						**Unutma Talepleri**				
	Tam Unutma	**Yaklaşık Unutma**	**Eksiksizlik / [illegible]**	**Zamanındalık / Hız**	**Doğruluk**	**Hafiflik**	**Kanıtlanabilir Garantiler**	**Doğrulanabilirlik**	**Öğe Çıkarımı**	**Özellik Çıkarımı**	**Sınıf Çıkarımı**	**Görev Çıkarımı**	**Akış Çıkarımı**
Modelden Bağımsız													
Diferansiyel gizlilik (Differential privacy) [36], [37]	✗	✓	✓	✓	-	✓	✓	-	✓	✗	✗	✗	✓
Sertifikalı kaldırma (Certified removal) [38], [28], [39], [40]	-	✓	-	✓	✓	✓	✓	-	✓	✗	✗	✗	✓
İstatistiksel sorgu öğrenimi (Statistical query learning)	-	✓	✓	✓	-	✓	-	-	✓	✗	✗	✗	-

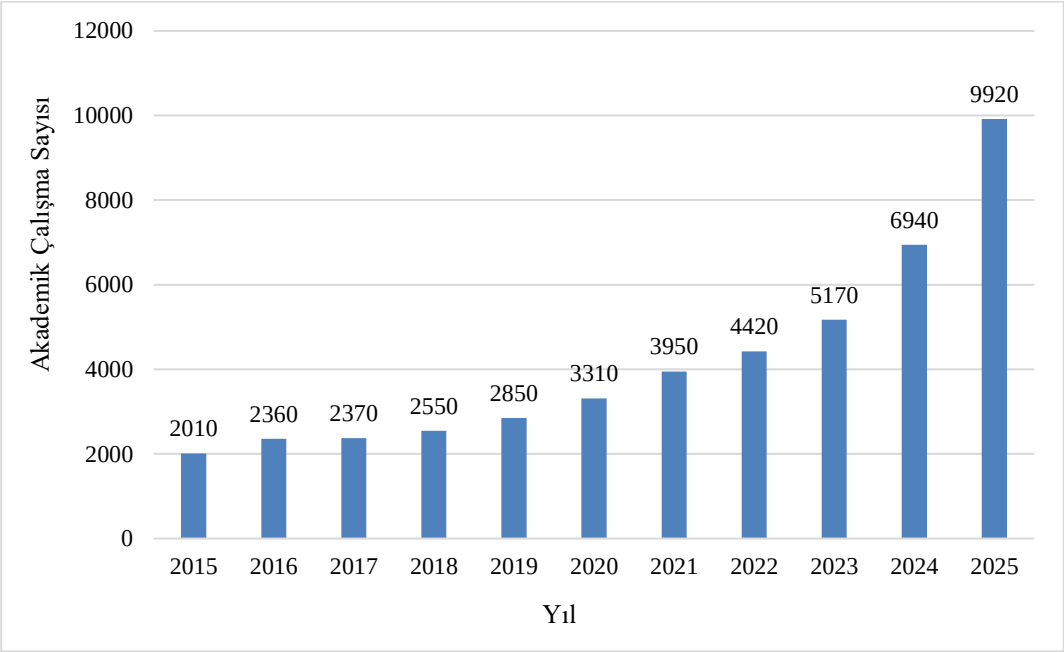
Unutma Yöntemleri	Unutma Tasarımı		Tasarım Gereksinimleri						Unutma Talepleri				
	Tam Unutma	Yaklaşık Unutma	Eksiksizlik / [illegible]	Zamanındalık / Hız	Doğruluk	Hafiflik	Kanıtlanabilir Garantiler	Doğrulanabilirlik	Öğe Çıkarımı	Özellik Çıkarımı	Sınıf Çıkarımı	Görev Çıkarımı	Akış Çıkarımı
[24]													
Azalan öğrenme (Decremental learning) [41], [31]	✗	-	✗	✓	✓	-	-	-	✓	✗	✗	✗	✗
Bilgi adaptasyonu (Knowledge adaptation) [42], [43], [44], [45]	✗	✓	-	-	-	✗	✗	-	✓	-	-	-	-
Parametre örnekleme (Parameter sampling) [46]	✗	✓	-	-	-	-	-	-	✓	✗	✗	✗	✗
Modele Özgü													
Softmax sınıflandırıcılar (Softmax classifiers) [47]	✗	✓	✓	-	-	✓	✗	✓	✗	✗	✓	✗	✗
Doğrusal modeller (Linear models) [48], [49]	✓	✗	-	✓	-	✓	✓	✓	✓	✗	✗	✗	✗

Unutma Yöntemleri	Unutma Tasarımı		Tasarım Gereksinimleri						Unutma Talepleri				
	Tam Unutma	Yaklaşık Unutma	Eksiksizlik / [illegible]	Zamanındalık / Hız	Doğruluk	Hafiflik	Kanıtlanabilir Garantiler	Doğrulanabilirlik	Öğe Çıkarımı	Özellik Çıkarımı	Sınıf Çıkarımı	Görev Çıkarımı	Akış Çıkarımı
Ağaç tabanlı modeller (Tree-based models) [50], [51]	✗	-	✗	-	✓	✗	✗	✗	✓	✗	✗	✗	✗
Bayes tabanlı modeller (Bayesian models) [46]	-	✓	-	-	✓	-	✓	-	✓	✗	✗	✗	✗
YSA tabanlı modeller (DNN-based models) [52], [27], [53], [54], [55], [56], [57]	✗	✓	-	-	✓	-	-	-	✓	✗	✗	✗	✗
Veri Odaklı													
Veri bölmeleme (Data partition) [58], [25], [59]	✓	✗	✓	✗	-	✗	✓	✓	✓	✗	-	✗	-
Veri artırma (Data augmentation) [60], [61], [62], [63]	✗	✓	-	-	-	-	✗	-	-	✗	✓	✗	✗

Unutma Yöntemleri	Unutma Tasarımı		Tasarım Gereksinimleri						Unutma Talepleri				
	Tam Unutma	Yaklaşık Unutma	Eksiksizlik / [illegible]	Zamanındalık / Hız	Doğruluk	Hafiflik	Kanıtlanabilir Garantiler	Doğrulanabilirlik	Öğe Çıkarımı	Özellik Çıkarımı	Sınıf Çıkarımı	Görev Çıkarımı	Akış Çıkarımı
Veri etkisi (Data influence) [64], [65], [66]	✗	✓	-	✓	-	✓	✓	-	✓	✗	✗	✗	✗

Tablo 2.3. incelendiğinde, mevcut literatürün ağırlıklı olarak tekil veri örneklerinin silinmesine odaklandığı, modelin belirli bir veri sınıfını veya yapısal özelliğini unutması gerektiğinde ise mevcut yöntemlerin yetersiz kaldığı görülmektedir. Geleneksel derin öğrenme modellerinin bu karmaşık unutma taleplerini karşılayamaması ve doğrulanabilirlik garantisi sunamaması, yapısal bir sınır olarak öne çıkmaktadır. Literatürdeki bu boşluk, veriler arasındaki yapısal bağları açıkça modelleyebilen ilişkisel sınıflandırma tabanlı bir yaklaşımın gerekliliğini ortaya koymaktadır. Bu doğrultuda tez çalışması kapsamında geliştirilen FedFORGET yöntemi, verinin yalnızca kendisini değil aynı zamanda ilişkisel bağlamını da hedef alarak Tablo 2.3. ile vurgulanan sınıf/özellik unutması ve doğrulanabilirlik eksikliklerine yapısal bir çözüm sunmayı amaçlamaktadır.

MU alanındaki araştırma eğilimini görselleştirmek ve literatürün genel gelişim seyrini değerlendirmek amacıyla, konuya ilişkin akademik çalışmaların nicel bir incelemesi Şekil 2.3. ile sunulmuştur. İlgili grafik, 2015–2025 yılları arasında Google Scholar veri tabanında "Machine unlearning" anahtar kelimesi kullanılarak elde edilen sonuçların yıllara göre dağılımını göstermektedir. Bu dağılım, içerik uygunluğu açısından detaylı bir filtreleme yapılmaksızın oluşturulduğundan, kesin bir literatür analizi sunmaktan ziyade alanın genel araştırma eğilimini yansıtmayı amaçlamaktadır. Şekil incelendiğinde, makine unutmasına yönelik akademik ilginin yıllar içinde istikrarlı biçimde arttığı ve yayımlanan çalışma sayısında belirgin bir yükseliş eğilimi olduğu görülmektedir.



Şekil 2.3. "Machine unlearning" anahtar kelimesi kullanılarak Google Scholar veri tabanında yapılan arama sonuçlarının 2015–2025 yılları arasındaki yıllara göre dağılımı.

MU alanında güncel yaklaşımları ve bu yaklaşımların pratik uygulanabilirliğini değerlendirmek amacıyla literatürdeki öne çıkan açık kaynaklı projeler incelenmiştir. Bu bağlamda, araştırma topluluğu tarafından geliştirilen ve genel erişime açık kod depolarıyla desteklenen başlıca MU algoritmaları, hedeflenen model mimarilerine göre kategorize edilerek Tablo 2.4. ile sunulmuştur. Bu tablo, teorik olarak geliştirilen MU yaklaşımlarının, hangi makine öğrenmesi mimarileri (Derin Öğrenme, Graf Modelleri, Federe Öğrenme, Büyük Dil Modelleri vb.) üzerinde pratikte hayat bulduğunu göstermesi açısından referans niteliği taşımaktadır.

Tablo 2.4. Literatürde geliştirilen ve açık kaynak olarak sunulan makine unutması algoritmaları.

Unutma Algoritmaları ve Referansları	**Uygulanabilir Makine Öğrenmesi Modelleri**	**Kod Deposu**
SISA [25]	Modelden bağımsız	https://github.com/cleverhans-lab/machine-unlearning
Athena [67], [68]	Modelden bağımsız	https://github.com/inspire-group/unlearning-verification
AmnesiacML [28]	Modelden bağımsız	https://github.com/lmgraves/AmnesiacML
Kpriors [69]	Modelden bağımsız	https://github.com/team-approx-bayes/kpriors

Unutma Algoritmaları ve Referansları	Uygulanabilir Makine Öğrenmesi Modelleri	Kod Deposu
ERM [39]	Modelden bağımsız	https://github.com/ChrisWaites/descent-to-delete
ShallowAttack [70]	Modelden bağımsız	https://github.com/MinChen00/UnlearningLeaks
UnrollingSGD [71]	Modelden bağımsız	https://github.com/cleverhans-lab/unrolling-sgd
DeltaGrad [72]	Modelden bağımsız	https://github.com/thuwuyinjun/DeltaGrad
Amnesia [73]	Modelden bağımsız	https://github.com/schelterlabs/projects-amnesia
SCAR [74]	Modelden bağımsız	https://github.com/jbonato1/SCAR
KGA [44]	Modelden bağımsız	https://github.com/Lingzhi-WANG/KGAUnlearn
MUPy [24]	KNN	https://github.com/theLauA/MachineUnlearningPy
DelKMeans [31]	k-ortalama algoritması	https://github.com/tginart/deletion-efficient-kmeans
CertifiedRem [75]	Doğrusal modeller	https://github.com/facebookresearch/certified-removal
CertAttack [76]	Doğrusal modeller	https://github.com/ngmarchant/attack-unlearning
PRU [48]	Doğrusal modeller	https://github.com/zleizzo/datadeletion
DeltaBoost [51]	Ağaç tabanlı modeller	https://github.com/Xtra-Computing/DeltaBoost
HedgeCut [50]	Ağaç tabanlı modeller	https://github.com/schelterlabs/hedgecut
DaRE-RF [77]	Ağaç tabanlı modeller	https://github.com/jjbrophy47/dare_rf
MCMC-Unlearning [78]	Bayes tabanlı modeller	https://github.com/fshp971/mcmc-unlearning
BIF [79]	Bayes tabanlı modeller	https://github.com/fshp971/BIF
L-CODEC [56]	Derin öğrenme	https://github.com/vsingh-group/LCODEC-deep-unlearning
Selective Forgetting [38], [53]	Derin öğrenme	https://github.com/AdityaGolatkar/SelectiveForgetting
Neurons [80]	Derin öğrenme	https://github.com/Hunter-DDM/knowledge-neurons
Unlearnable [60]	Derin öğrenme	https://github.com/HanxunH/Unlearnable-Examples
DLMA [63]	Derin öğrenme	https://github.com/AnonymousDLMA/MI_with_DA
ERM-KTP [81]	Derin öğrenme	https://github.com/RUIYUN-ML/ERM-KTP

Unutma Algoritmaları ve Referansları	Uygulanabilir Makine Öğrenmesi Modelleri	Kod Deposu
GraphProjector [82]	Graf modelleri	https://github.com/CongWeilin/Projector
GraphEditor [83]	Graf modelleri	https://anonymous.4open.science/r/GraphEditor-NeurIPS22-856E/README.md
FedLU [84]	Graf modelleri	https://github.com/nju-websoft/FedLU/
GUIDE [85]	Graf modelleri	https://github.com/Happy2Git/GUIDE
GNNDelete [86]	Graf modelleri	https://github.com/mims-harvard/GNNDelete
GraphEraser [87]	Graf modelleri	https://github.com/MinChen00/Graph-Unlearning
GST-Unlearn [88]	Graf modelleri	https://zenodo.org/records/7613150
RecEraser [89]	Tavsiye sistemleri	https://github.com/chenchongthu/Recommendation-Unlearning
ADV-MULTVAE [90]	Tavsiye sistemleri	https://github.com/CPJKU/adv-multvae
FedEraser [91]	Federe öğrenme	https://www.dropbox.com/s/1lhx962axovbbom/FedEraser-Code.zip?dl=0
RapidFed [13]	Federe öğrenme	https://github.com/yiliucs/federated-unlearning
SIFU [37]	Federe öğrenme	https://github.com/Accenture/Labs-Federated-Learning/tree/SIFU
Fast-FedUL [92]	Federe öğrenme	https://github.com/thanhtrunghuynh93/fastFedUL
FATS [93]	Federe öğrenme	https://github.com/Happy2Git/FATS_supplement
EUL [94]	Büyük dil modelleri	https://github.com/SALT-NLP/Efficient_Unlearning
E2URec [95]	Büyük dil modelleri, Tavsiye sistemleri	https://github.com/justarter/E2URec
Ext-Sub [96]	Büyük dil modelleri	https://github.com/HITsz-TMG/Ext-Sub
SP [97]	Büyük dil modelleri	https://github.com/nickypro/selective-pruning
Quark [98]	Büyük dil modelleri	https://github.com/GXimingLu/Quark
I2I-Unlearn [99]	Üretken modeller	https://github.com/jpmorganchase/i2i-generator-unlearning
AdvUnlearn [100]	Üretken modeller	https://github.com/OPTML-Group/AdvUnlearn
FAST [101]	Üretken modeller	https://github.com/Subhodip123/weak-unlearning-gan

MU alanındaki teorik çerçevenin yanı sıra, bu teknolojinin mevcut literatürdeki uygulama pratiklerinin kapsamlı bir analizi de büyük önem taşımaktadır. İlgili literatür incelemesi kapsamında makine unutmasının gerçek dünyadaki uygulama alanları, sistemin zafiyetlerini hedef alan makine unutmasına karşı saldırılar ve bu tehditleri önlemeye yönelik mahremiyeti koruyan teknikler detaylı biçimde incelenmiştir. Bu bağlamda literatürdeki mevcut çalışmalar, odaklandıkları gerçek dünya senaryolarına göre sistematik biçimde kategorize edilmiş ve bu çalışmalara ait sınıflandırma Tablo 2.5. ile sunulmuştur. İlgili literatür incelendiğinde araştırmaların ağırlıklı olarak mobil uygulamalarda unutma, IoT sistemlerindeki yaklaşımlar, kurumsal sistemler, görsel ve büyük dil modelleri, kişiselleştirilmiş tavsiye sistemleri ve sağlık ile finans gibi diğer özel alanlar üzerinde yoğunlaştığı görülmektedir.

Tablo 2.5. Makine unutması çalışmalarının uygulama alanlarına göre sınıflandırılması.

Çalışma Alanı	Referans	Özet Açıklama
Mobil Uygulamalar	[102]	Bu çalışmada, mobil ve kaynak kısıtlı ortamlarda sınıf bazlı veri silme işlemini etkin şekilde gerçekleştirmek amacıyla, ek veri gerektirmeyen ve knowledge distillation kullanan CLUE adlı düşük gecikmeli ve enerji verimli bir makine unutma yöntemi önerilmiştir.
IoT Sistemleri	[103]	Veri akışlarında meydana gelen concept drift problemini ele alan bu çalışmada, yeniden eğitim gerektirmeden drift etkisini ortadan kaldırmayı amaçlayan, autoencoder tabanlı ve modelden bağımsız bir unlearning yaklaşımı önerilmiştir. Yöntem, özellikle su dağıtım ağları gibi gerçek zamanlı IoT sistemlerinde ve görüntü tabanlı uygulamalarda test edilmiştir.
	[104]	Bu çalışmada, IoT tabanlı FL sistemlerinde zararlı istemcilerin neden olduğu hatalı model güncellemelerini tespit ederek unlearning işlemiyle modelden çıkaran ve düşük maliyetli, gerçek zamanlı çalışan FedRemover yöntemi önerilmiştir.
Kurumsal Sistemler	[105]	Bu çalışmada, kullanıcı hareket verilerinin gizliliğini sağlamak amacıyla, yeniden eğitime gerek kalmadan belirli kullanıcı trajektorilerini modelden silen ve önem tabanlı örnek seçimi ile teacher–student yaklaşımını birleştiren TraceHiding adlı ölçeklenebilir bir makine unutma yöntemi

		önerilmiştir.
Tavsiye Sistemleri	[106]	Bu çalışmada, web servis seçiminde kullanılan QoS tahmin modelleri için, veri gizliliği ve güvenlik gereksinimlerini karşılamak amacıyla yeniden eğitime gerek kalmadan etkili veri silme sağlayan QoSEraser adlı makine unutma tabanlı bir çerçeve önerilmiştir.
	[107]	Bu çalışmada, akıllı şehirlerde veri gizliliğini sağlamak amacıyla, yeniden eğitime gerek kalmadan hızlı veri silme gerçekleştiren ve GAN tabanlı adversarial öğrenme yaklaşımını kullanan bir makine unutma yöntemi önerilmiştir.
Görüntü ve Doğal Dil İşleme Uygulamaları	[108]	Bu çalışmada, bilinmeyen ortamlarda nesne tanıma problemini ele almak amacıyla, yalnızca 3D nokta bulutu verisi kullanarak gerçek zamanlı topolojik haritalama yapan ve uzak nesnelerde veri yoğunluğunu artıran AiS-GNG yöntemi önerilmiştir.
	[109]	Bu çalışmada, büyük dil modellerinde (LLM) istenmeyen veya hassas bilgilerin modelden çıkarılmasını ele alan makine unutma yaklaşımları kavramsal, metodolojik ve değerlendirme boyutlarıyla incelenmiş ve güvenli, güvenilir yapay zeka sistemleri için bir çerçeve sunulmuştur.
Saldırı Tespiti	[110]	Bu çalışmada, saldırı tespit sistemlerinin belirli saldırı türlerini yeniden eğitime gerek kalmadan unutmasını sağlayan ve knowledge distillation tabanlı MT-KD yöntemi önerilmiştir.
	[111]	Bu çalışmada, anomali tespiti için yeniden eğitime gerek kalmadan hızlı model güncellemesi sağlayan ve verileri sanal istemciler olarak ele alan ViFLa adlı federated learning tabanlı makine unutma yaklaşımı önerilmiştir.

MU, verileri silerek mahremiyeti korumayı amaçlasa dahi model üzerinde yeni güvenlik açıkları ortaya çıkabilmektedir. Bu saldırılar mahremiyet saldırıları ve güvenlik saldırıları olmak üzere temel olarak iki ana gruba ayrılır.

Mahremiyet saldırıları, unutma işleminden sonra, silindiği sanılan verilerin izlerini sürerek bilgi çalmayı hedefler. Aşağıda mahremiyet saldırılarına sunulan örnekler yer almaktadır.

Üyelik Çıkarım Saldırısı (Membership Inference Attack - MIA): Hedef verinin eğitim setinde yer alıp almadığını tespit etmeyi amaçlar. Unutma işlemi uygulanmış modeller, standart modellere kıyasla daha yüksek bilgi sızıntısı riski barındırır [70]. Model çıktılarının yalnızca etiket olarak paylaşılması dahi mutlak güvenlik sağlamamakta küçük manipülasyonlarla silinen veriler açığa çıkarılabilmektedir [112].

Veri Yeniden Yapılandırma Saldırısı (Data Reconstruction Attack - DRA) ve Tersine Çevirme: Saldırganlar, modelin unutma öncesi ve sonrası çıktıları arasındaki istatistiksel farkları analiz ederek sistemden silindiği varsayılan verinin özniteliklerini ve etiketlerini geri dönüştürüp kurtarabilmektedir [113].

Gölge Model Saldırıları (Shadow Model Attacks): Saldırganın, hedef modelin dışarıya sunduğu çıktıları analiz ederek sistemin davranışını taklit eden kopya modeller üretmesidir. Böylece saldırgan, asıl sisteme doğrudan erişemese bile kendi ürettiği kopyalar üzerinden eğitim verilerini tahmin edip ele geçirebilir [70].

Güvenlik Saldırıları, modelin unutma işlemi yaparken performansını veya karar verme yeteneğini bozmayı hedefler. Aşağıda güvenlik saldırılarına sunulan örnekler yer almaktadır.

Veri Zehirleme Saldırısı (Data Poisoning Attack) ve Kötü Amaçlı Unutma Talebi (Malicious Unlearning Request Attack): Eğitim aşamasında sisteme kasıtlı olarak zehirli verilerin sızdırılması [114] veya asılsız veri silme taleplerinin iletilmesi [115] yöntemleridir. Bu saldırılarda temel amaç, unutma işlemi esnasında modelin istikrarını bozarak genel tahmin doğruluğunu ve performansını zayıflatmaktır.

Sistem Kırma Saldırıları (Jailbreak Attack): Özellikle büyük dil modellerinde güvenlik filtrelerini aşmak üzere tasarlanmış komutlar kullanılarak modelin silindiği varsayılan veya gizlemesi gereken hassas bilgileri ifşa etmesini hedeflemektedir [116].

MU süreci, sadece hedef verinin eğitim setinden fiziksel olarak silinmesi işlemi olarak değerlendirilemez. Buradaki temel amaç, ilgili verinin modelin parametreleri ve karar mekanizmaları üzerindeki tüm istatistiksel izlerinin yasal düzenlemelere uygun bir biçimde ortadan kaldırılmasıdır. Bu süreçte, hedeflenen verinin etkileri silinirken modelin genel doğruluğunun, kararlılığının ve performansının bozulmaması kritik bir öneme sahiptir. Bu dengeyi kurmak ve olası bilgi sızıntılarını engellemek amacıyla literatürde çeşitli mahremiyet koruyucu tekniklerinden faydalanılmaktadır.

Diferansiyel Mahremiyet (Differential Privacy): Eğitim verilerinin arasına kasten bir miktar rastgele karmaşa ekleyerek, bireylerin kimliğini kalabalığın içinde görünmez kılma yöntemidir. Model genel eğilimleri anlar ancak çözemez. Sistemdeki asıl zorluk, bu karmaşanın miktarını dengelemektir. Verileri saklamak adına sisteme gereğinden fazla karmaşa yüklenirse, modelin öğrenme yeteneği bozulur ve ürettiği sonuçlar işe yaramaz hale gelir [117].

Homomorfik Şifreleme (Homomorphic Encryption): Verilerin şifresini hiç açmadan üzerlerinde matematiksel hesaplamalar yapmaya olanak tanıyan bir güvenlik tekniğidir. Sistemdeki veriler en üst düzeyde korunmaktadır. Fakat her şey şifreli olduğu için modelin içinde kimin ne kadar payı olduğu görünmez hale gelir. Bu aşırı gizlilik, bir noktada dezavantaja dönüşür. Hedefe yönelik bir unutma işlemini uygulamayı zorlaştırır [118].

Güvenli Çok Taraflı Hesaplama (Secure Multi-party Computation): Merkezi bir yöneticiye veya ortak bir veri havuzuna ihtiyaç duymadan, birden fazla tarafın kendi kapalı sistemleri içinde kalarak ortak bir model geliştirmesine imkan tanımaktadır. Hiçbir kurum diğerinin verisini görmediği için gizlilik ihlali riski ortadan kalkar. Sistemler arası sürekli iletişim gerektirmesi nedeniyle ağ maliyetleri yüksektir [119].

Güvenilir Çalıştırma Ortamları (Trusted Execution Environments - TEE): Bu yöntem ile işlemci seviyesinde donanımsal olarak korunan, dış müdahalelere tamamen kapalı ve izole bir çalışma alanı oluşturulur [120].

Gürültü Ekleme ve Gizleme (Noise Addition & Obfuscation): Algoritmanın matematiksel işlemlerine rastgele istatistiksel gürültüler katarak, bireysel veri noktalarının model üzerindeki belirgin etkisini maskeler [121].

Erişim Kontrolü ve Denetim İzleri (Access Control & Audit Trails): Mahremiyetin yalnızca matematiksel algoritmalarla değil, idari güvenlik politikalarıyla da desteklenmesi yaklaşımıdır. Kötü niyetli silme taleplerinin önüne geçmek amacıyla kimlik doğrulama mekanizmaları ve sistemdeki tüm işlemlerin geriye dönük olarak incelenebilmesini sağlayan şeffaf kayıt sistemleri kullanılır [122] .

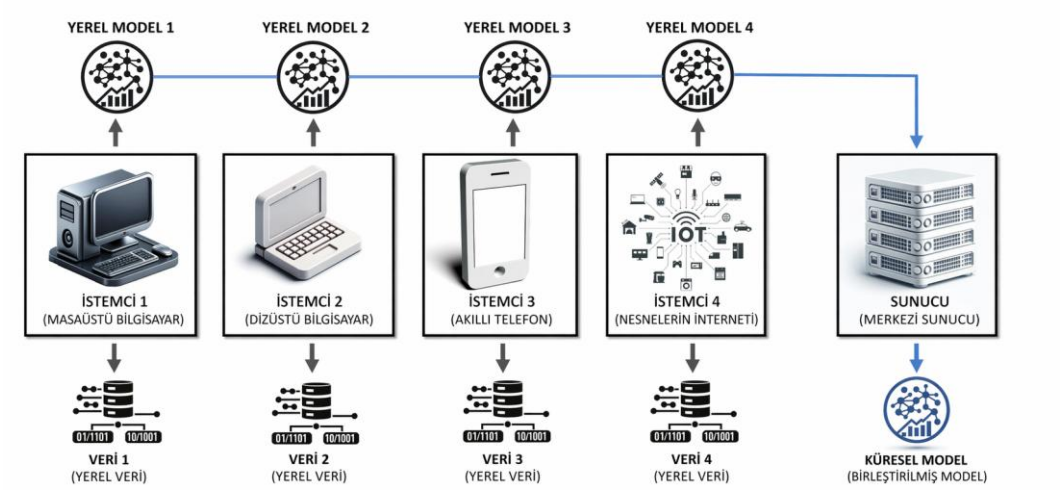
Bu bölümde, MU kavramı, temel yaklaşımları ve uygulama alanları detaylı bir şekilde incelenmiştir. Mevcut yöntemlerin güçlü ve sınırlı yönleri ele alınarak, özellikle veri gizliliği ve “unutulma hakkı” kapsamında taşıdığı önem vurgulanmıştır.

2.2. Federe Öğrenme (Federated Learning)

Günümüzde, internet tabanlı teknolojilerin hızla ilerlemesiyle birlikte, bu teknolojilerin kullanıcıları sürekli olarak veri üretmektedir [123]. Üretilen bu veriler işlenerek daha anlamlı hale getirilmektedir. Güvenlik, eğitim, sağlık, finans vb. alanlarda bu anlamlı verileri kullanan devletler, kurumlar veya kuruluşlar, mevcut durumu daha iyi analiz edebilir ve geleceği daha isabetli bir şekilde öngörebilirler.

Öte yandan, kişisel bilgiler içeren bu üretilmiş veriler mahremiyet açısından büyük önem taşımaktadır [124]. Verilerin sunucularda depolanması ve paylaşılması, veri mahremiyetini korumaya yönelik çeşitli güvenlik sorunlarını da beraberinde getirmektedir. Ülkeler, veri mahremiyeti ve veri güvenliği ihtiyaçlarını karşılamak amacıyla KVKK, GDPR gibi yasal düzenlemeleri hayata geçirmişlerdir. Ancak, güvenliği sağlamak için yalnızca yasal düzenlemeler ile hareket etmek yeterli değildir. Gerçek bir güvenliğin tesis edilebilmesi için teknolojik çözümlere de ihtiyaç duyulmaktadır [125].

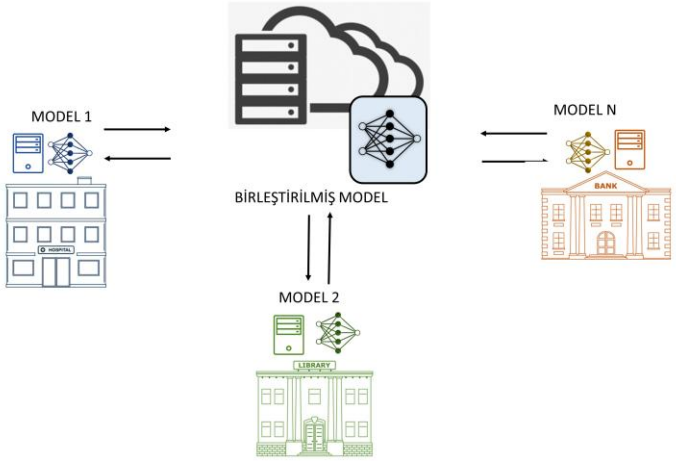
Veri mahremiyetini sağlayarak güvenlik endişelerini gidermek amacıyla 2016 yılında Google tarafından geliştirilen federe öğrenme, yeni nesil bir yapay zeka teknolojisidir [126]. FL, istemciler ve sunuculardan oluşan dağıtık sistemler için tasarlanmış bir teknolojidir [127], [128]. Bu teknoloji sayesinde, istemciler tarafından üretilen veriler sunucuya iletilmez. Bunun yerine, her bir istemcide üretilen veriler kullanılarak yerel bir model eğitilir ve ardından eğitilen bu modeller sunucuya aktarılır. Sunucuya iletilen modeller, merkezi bir model oluşturmak üzere birleştirilir. Oluşturulan bu merkezi model daha sonra istemcilere geri dağıtılır [129]. Şekil 2.4. FL mimarisinin nasıl çalıştığını göstermektedir.



Şekil 2.4. Federe öğrenme mimarisi.

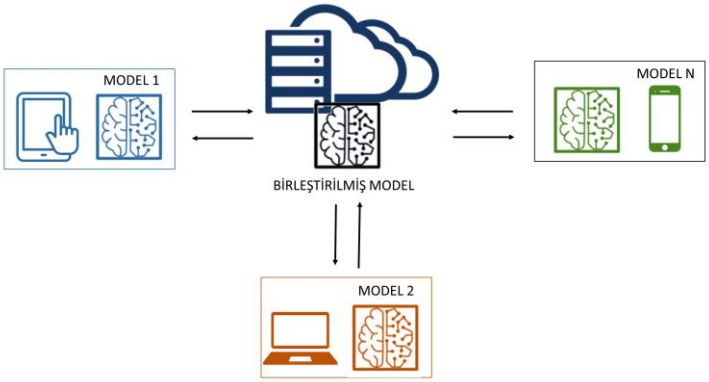
İstemci türlerine ve ürettikleri verilere bağlı olarak FL çapraz-silo (cross-silo) ve çapraz-cihaz (cross-device) olmak üzere iki farklı türe ayrılmaktadır [130].

Çapraz-silo FL, istemcilerin bankalar, hastaneler vb. şirketler veya kurumsal organizasyonlardan oluştuğu bir mimaridir. Bu mimari istemci sayısının sınırlı olduğu, verilerin oldukça hassas nitelik taşıdığı ve farklı istemciler arasında kolayca paylaşılamadığı durumlarda kullanılır [130], [131]. Şekil 2.5. müşterileri bankalar olan bir çapraz-silo FL mimarisi örneğini sunmaktadır.



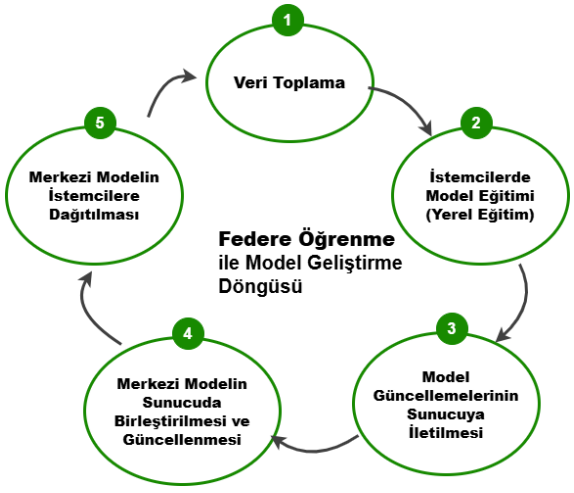
Şekil 2.5. Çapraz-silo federe öğrenme**.**

Çapraz-cihaz federe öğrenme, istemcilerin bilgisayarlar, tabletler, akıllı telefonlar, giyilebilir cihazlar vb. teknolojik aletlerden oluştuğu mimaridir. Bu yapıda istemci sayısı çok daha fazladır [132], [133]. Mobil cihaz uygulamaları bu mimariye örnek olarak gösterilebilir. Şekil 2.6. çapraz-cihaz FL mimarisini göstermektedir.



Şekil 2.6.Çapraz-cihaz federe öğrenme.

FL sistemlerinde, merkezi tasarım (centralized design) [134], [135] ve merkeziyetsiz tasarım (decentralized design) [136], [137] olmak üzere iki temel iletişim yöntemi bulunmaktadır. Merkezi tasarımda, istemciler üzerinde eğitilen model parametreleri doğrudan sunucuya iletilir. Güncellemeler sunucu üzerinde işlenir ve istemcilere geri gönderilir [138]. Sunucu ile istemciler arasındaki bu iletişim eş zamanlı [139] veya eş zamanlı olmayan [140] yapıda olabilir. Merkeziyetsiz tasarımda ise iletişim doğrudan istemciler arasında gerçekleşir ve her bir istemci merkezi model parametrelerini anında güncelleyebilir [141]. Günümüzde merkezi tasarım çok daha yaygın kullanılmaktadır. Ancak, bilgilerin tek bir sunucuda toplanmasının doğurabileceği potansiyel riskler nedeniyle merkeziyetsiz tasarım da tercih edilmektedir. Ne var ki, merkeziyetsiz bir iletişim mimarisi tasarlamak son derece zorlayıcı bir süreçtir.



Şekil 2.7. Federe öğrenme ile model geliştirme süreci.

Literatürde FL mimarisi için yaygın olarak kullanılan merkezi tasarımda model eğitim adımları Şekil 2.7. ile açıklanmaktadır. Bu adımlar, model eğitiminin ilk döngüsü tamamlandıktan sonra; ya model önceden hedeflenen bir performans seviyesine ulaşana kadar ya da sistem tasarımcısı tarafından belirlenen toplam iletişim turu sayısı tamamlanana kadar devam etmektedir. Modelin eğitim süreci temel olarak beş aşamadan oluşmaktadır [8]:

1. Veri Toplama: Her bir uç cihaz / istemci, kendi donanımı üzerinde yerel veriler üretir ve yalnızca bu veriler üzerinde işlem yapar. Üretilen bu veriler, mahremiyet ilkesi gereği kesinlikle merkezi bir sunucuya aktarılmaz. Cihazın kendi içinde güvenle tutulur.

2. İstemcilerde Model Eğitimi (Yerel Eğitim): Her bir istemci, kendi ürettiği verileri kullanarak yerel bir öğrenme süreci yürütür ve model eğitir. İlerleyen turlarda, sunucudan en güncel merkezi model istemciye indirilir. Bu noktadan sonra istemci ya sadece bu modeli çalıştırıp kullanmaya devam eder ya da modelin eğitimini sürdürür. Bu eğitim stratejisi, istemcinin donanım kapasitesine göre şu iki yöntemden biriyle gerçekleştirilir:

- **Artımlı Öğrenme (Incremental Learning):** Cihazda o an üretilen yeni veriler, modelin mevcut bilgi birikiminin üzerine eklenerek eğitime devam edilir [142].

- **Mevcut Model ile Eş Zamanlı Eğitim:** İstemciye iletilen merkezi model, istemci üzerinde aktif çalışmayı sürdürürken, eş zamanlı olarak cihazda üretilen yeni veriler kullanılarak yerel bir model daha eğitilir. Eğitimi tamamlanan bu yerel modelin parametreleri, birleştirme işlemi için merkezi sunucuya iletilir. Bu mimari sayesinde, uç cihazlarda mevcut global modelin operasyonel kullanımı devam ederken sunucu tarafında ağdaki tüm cihazlardan gelen güncel parametrelerin sürekli olarak birleştirilmesiyle merkezi modelin iyileştirilmesi ve güncellenmesi süreci paralelde yürütülmüş olur.

3. Model Güncellemelerinin Sunucuya İletilmesi: İstemcilerde eğitilen modelin ulaştığı yeni parametreler, her bir istemciden merkezi sunucuya iletilir. Ağa yalnızca bu matematiksel parametreler çıktığı için veri gizliliği korunmuş olur.

4. Merkezi Modelin Sunucuda Birleştirilmesi ve Güncellenmesi: Ağa bağlı tüm istemcilerden gelen yerel model güncellemeleri merkezi sunucuda toplanır. Ardından bu güncellemeler, federe birleştirme (federe aggregation) algoritmaları kullanılarak tek bir yapı altında harmanlanır ve yeni merkezi model elde edilir. Sonraki her turda, istemcilerden

gelen yeni güncellemeler yine bu adımda birleştirilerek merkezi modelin sürekli daha yetenekli hale gelmesi sağlanır.

5. Merkezi Modelin İstemcilere Dağıtılması: Sunucuda elde edilen güncel merkezi model, ağdaki tüm istemcilere geri gönderilir. Bu adımın tamamlanmasıyla birlikte model eğitiminin ilk döngüsü sona ermiş olur. Bu işlemler bütünü, hedeflenen tur sayısına ulaşılana dek tekrarlanır.

FL sistemlerinde, farklı istemcilerde eğitilen yerel modellerden elde edilen parametrelerin bir araya getirilerek merkezi modelin güncellenmesi işlemine birleştirme (aggregation) adı verilmektedir. Federe birleştirme algoritmaları model performansını optimize etmek, iletişim maliyetlerini azaltmak ve veri mahremiyetini sağlamak amacıyla kullanılmaktadır [143]. İletişim mimarisi açısından değerlendirildiğinde merkezi federe öğrenmede birleştirme işlemi merkezi bir sunucu tarafından gerçekleştirilirken, merkeziyetsiz senaryolarda model entegrasyonu uçtan uca iletişim mekanizmalarıyla sağlanmaktadır.

Tablo 2.6. içerisinde literatüre kazandırılmış başlıca federe birleştirme algoritmaları, çalışma yaklaşımları ve uygulanabilirlik alanları özetlenmektedir. Bu tez çalışması kapsamında sınıflandırma kurallarına dayanan ve verilerin istatistiksel dağılımını kural setleri üzerinden harmanlayan FedCBA algoritması kullanılmaktadır. FedCBA, FL sistemlerinde cihazların ürettiği yerel sınıflandırma kurallarını destek ve güven ölçütlerine göre hazırlayarak merkezi ve açıklanabilir bir global model oluşturan kural tabanlı bir birleştirme algoritmasıdır [1].

Tablo 2.6. Federe öğrenme birleştirme algoritmaları.

Federe Birleştirme Algoritmaları	Açıklama	Kod Deposu
Federated Averaging (FedAvg) [144]	İstemcilerden gelen model ağırlıklarının, veri boyutuna oranla ağırlıklı ortalamasını alarak global modeli günceller.	https://github.com/KarhouTam/Federated-Averaging-PyTorch?utm_source=chatgpt.com
Federated Optimization with Proximal Term (FedProx) [145]	Temel yöntem olan FedAvg'nin heterojen veri dağılımlarında yaşadığı performans sorunlarına daha güçlü bir alternatif olarak geliştirilmiştir. Yerel modellerin eğitim esnasında merkezi modelden aşırı sapmasını engellemek amacıyla optimizasyon sürecine bir kısıtlama ekleyerek daha istikrarlı bir öğrenme süreci sunar.	https://github.com/litian96/FedProx

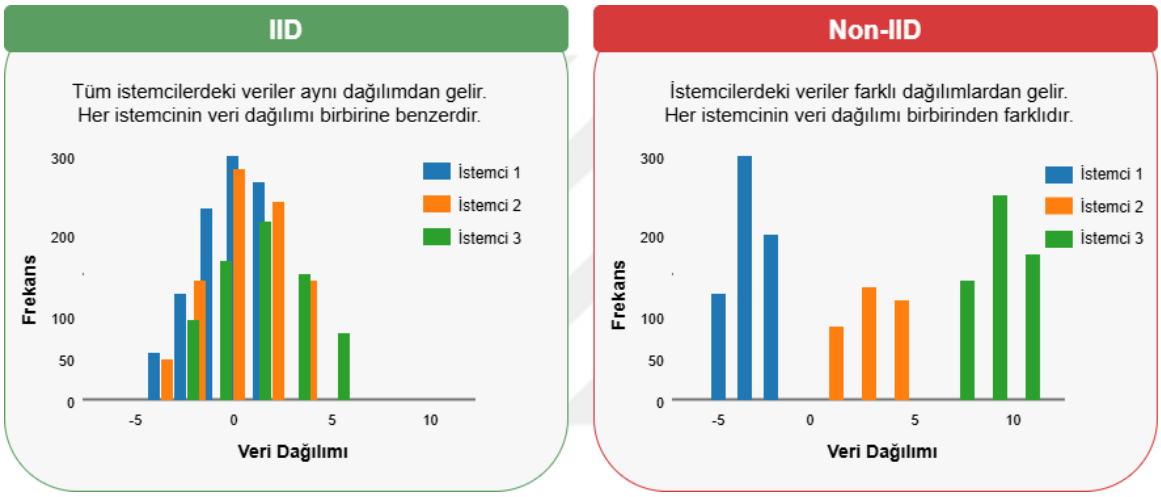
Federe Birleştirme Algoritmaları	Açıklama	Kod Deposu
Federated Matched Averaging (FedMA) [146]	Ağırlıkların doğrudan toplanması yerine, sinir ağlarındaki nöronları katman bazında eşleştirerek ortalamalarını alır.	https://github.com/IBM/FedMA
Federated Normalized Averaging (FedNova) [147]	İstemcilerin donanım farklılıkları nedeniyle attıkları farklı sayıdaki eğitim adımlarını normalize ederek adil birleştirme yapar.	https://github.com/JYWa/FedNova
Federated Bayesian Ensemble (FedBE) [148]	İstemcilerden gelen modellerin sadece ağırlıklarını ortalamak yerine bu modelleri Bayesçi çıkarım yöntemiyle bir topluluk olarak değerlendirir. Modelleri harmanlayarak, heterojen verilerde daha kararlı ve yüksek doğruluklu global bir model oluşturur.	https://github.com/sabhayjain/FedBE-TPMI-2021
Federated Personalization (FedPer) [149]	Modelin temel özellik çıkaran katmanlarını global olarak birleştirirken, sınıflandırıcı katmanlarını cihazda bırakarak kişiselleştirme sağlar.	https://github.com/ki-ljl/FedPer?utm_source=chatgpt.com
Federated Classification Based on Associations (FedCBA) [1]	Kural tabanlı CBA algoritmasını federe mimariyle ve mahremiyet kısıtlarıyla entegre eden geliştirilmiş birleştirme yöntemidir.	Bu tez çalışması kapsamında tasarlanmıştır.

Geleneksel merkezi makine öğrenmesi yaklaşımlarında, eğitim verisinin genellikle tek bir havuzda toplanarak homojen bir şekilde karıştırıldığı ve tek bir dağılımdan geldiği varsayılmaktadır. Buna karşılık, FL mimarisinde veri çoğunlukla istemcilerde yerel olarak tutulur ve işlenir; bu veri istemci tarafında üretilebileceği gibi, farklı kurumlardan doğrudan sisteme dahil edilen mevcut veri setleri de olabilir. Cihaz sahiplerinin farklı kullanım alışkanlıkları, coğrafi konumları ve demografik özellikleri nedeniyle, yerel verilerin istatistiksel dağılımları ağ genelinde büyük farklılıklar gösterir. Bu durum literatürde veri dağılımı problemi olarak ele alınır. İki ana kategoriye ayrılır:

1. Bağımsız ve Özdeş Dağılım (Independent and Identically Distributed - IID): [150] Makine öğrenmesi modellerinin eğitimi için istatistiksel olarak en ideal senaryoyu ifade eder. IID varsayımına göre ağdaki her bir istemcinin sahip olduğu yerel veri seti, tüm

sistemdeki global veri havuzunun kusursuz bir alt kümesi niteliğindedir. Yani, sistemdeki tüm sınıflar ve özellikler, her bir cihazda eşit ve orantılı olasılıklarla bulunur. FL'de IID dağılım sağlandığında, yerel modellerin gradyanları merkezi modele doğru tutarlı bir şekilde yakınsar ve modelin eğitim süreci yüksek doğrulukla, hızlı bir biçimde tamamlanır.

2. Bağımsız ve Özdeş Olmayan Dağılım (Non-Independent and Non-Identically Distributed - Non-IID): [150] Gerçek dünya FL uygulamalarında, istemcilerdeki veriler bağımsız ve özdeş dağılım göstermez. Kullanıcıların kendilerine özgü davranış kalıpları nedeniyle veriler yüksek oranda heterojendir. Non-IID durumlarında, istemciler kendi yerel ve yanlı verilerine aşırı uyum sağladıkları için, yerel model güncellemeleri global modelin genel performansını ve kararlılığını ciddi şekilde düşürür. Şekil 2.8. ile IID ve Non-IID durumlarına göre verinin istemciler arasında nasıl dağıldığı görsel olarak sunulmaktadır.



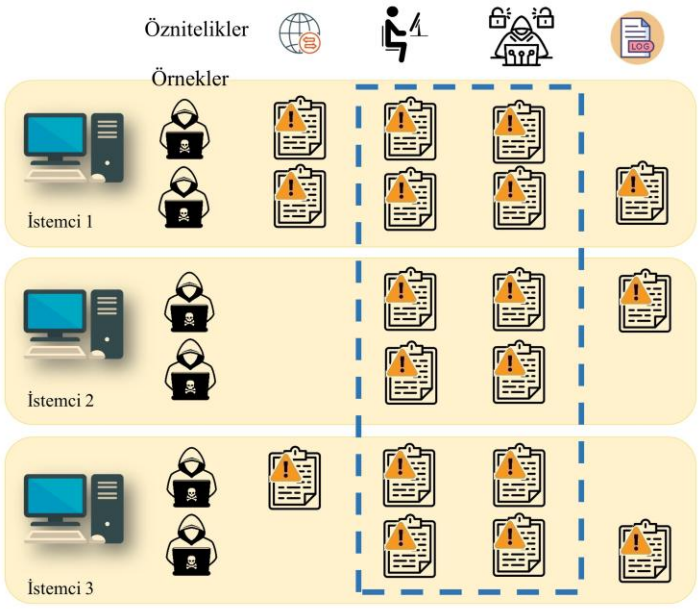
Şekil 2.8. IID ve Non-IID veri dağılımlarının istemciler üzerindeki karşılaştırmalı gösterimi.

FL mimarileri, ağdaki istemcilerin sahip olduğu yerel veri setlerinin özellikleri ve örneklemleri arasındaki kesişim durumlarına göre Yatay, Dikey ve Transfer Öğrenme olarak üç ana kategoriye ayrılmaktadır.

1. Yatay Federe Öğrenme (Horizontal Federated Learning - HFL) [151]: Yatay federe öğrenme, istemcilerdeki veri setlerinin özellik uzaylarının büyük ölçüde örtüştüğü, ancak örneklem uzaylarının farklı olduğu senaryoları kapsar. Bir diğer ifadeyle, veriler istemciler arasında yatay olarak bölünmüştür.

Şekil 2.9. HFL'ye örnek olarak verilmiştir. Şekle göre farklı coğrafi konumlarda bulunan ve farklı müşteri kitlelerine sahip olan kurumların ortak bir model eğitmesi bu kategoriye girer. Saldırı tespit sistemleri bağlamında ele alındığında, hedeflenen kurumlar veya

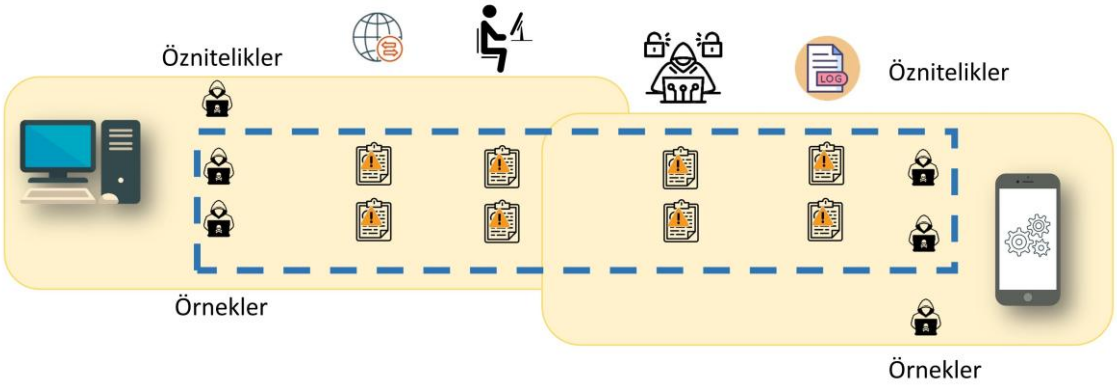
cihazlar farklı olsa da siber saldırıların karakteristik özellikleri benzerlik gösterir. HFL, bu ortak özellik uzayından faydalanarak ortak bir tespit modelinin eğitilmesini sağlar.



Şekil 2.9. Yatay federe öğrenme.

Dikey Federe Öğrenme (Vertical Federated Learning - VFL) [152]: Dikey federe öğrenme, veri setlerinin örneklem uzaylarının büyük ölçüde aynı olduğu, ancak özellik uzaylarının farklılık gösterdiği durumlarda kullanılır. Veriler kurumlar arasında dikey olarak bölünmüştür.

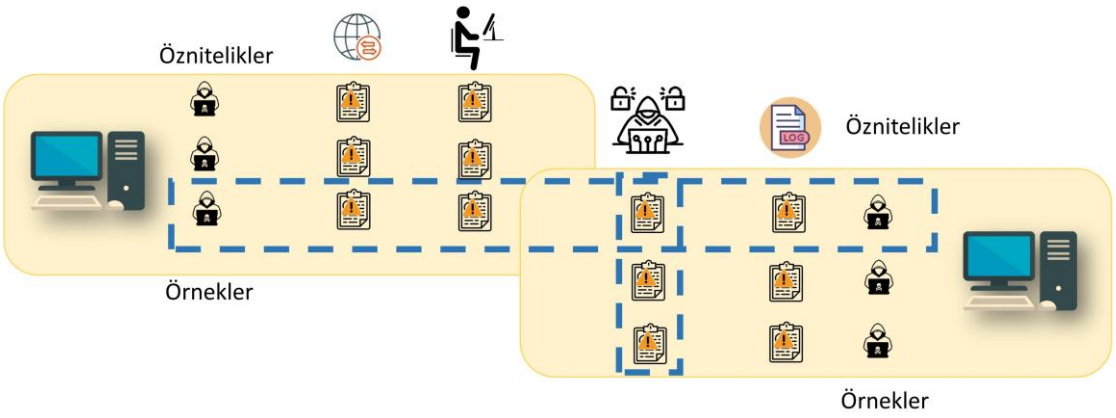
Şekil 2.10. VFL'ye örnek olarak verilmiştir. Şekle göre aynı kişilere ait farklı veri türlerinin birden fazla istemciye dağıldığı durumlar VFL için idealdir. Örneğin, kurumsal bir ağ üzerinde çalışan bir saldırı tespit sistemi, ağ trafiği ve potansiyel sızma girişimleri hakkında veri toplarken aynı kullanıcıların bireysel cihazlarına kurulan başka bir saldırı tespit sistemleri, cihaz kullanım alışkanlıkları ve uygulama davranışları hakkında veri toplar. Her iki istemci de aynı kullanıcılara hizmet verir, ancak topladıkları özellikler tamamen farklıdır. VFL mimarisi, bu iki istemciden gelen farklı özellikleri güvenli bir şekilde birleştirerek daha kapsamlı bir model eğitilmesine olanak tanır.



Şekil 2.10. Dikey federe öğrenme.

Federe Transfer Öğrenme (Federated Transfer Learning - FTL) [153]: Federe transfer öğrenme, istemcilerdeki veri setlerinin hem özellik uzaylarının hem de örneklem uzaylarının minimum düzeyde örtüştüğü veya tamamen farklı olduğu senaryolarda uygulanır. Veri setleri hem yatay hem de dikey olarak bölünmüş durumdadır.

Şekil 2.11. ile uygulama örneği sunulan FTL, veri miktarının veya etiketlerin yetersiz olduğu durumlarda organizasyonlar arası bilgi aktarımı için kullanılır. Örneğin, iki farklı ağ ortamı bulunduğunda daha geniş bir veri setine sahip olan ortamda eğitilen modelin parametreleri, veri boyutu kısıtlı olan ortama aktarılır. Aktarılan bu parametreler, kısıtlı veri kullanılarak optimize edilir. Bu yaklaşım, doğrudan veri paylaşımı yapılmaksızın etki alanları arasında bilgi transferi gerçekleştirilmesini sağlar.



Şekil 2.11. Federe transfer öğrenme.

FL mimarisi, veri işleme süreçlerine getirdiği yenilikçi yaklaşımlar sayesinde önemli avantajlar sunarken, dağıtık yapısından kaynaklanan bazı istatistiksel ve sistemsel zorlukları da beraberinde getirmektedir [154]. Bu durumlar Tablo 2.7. içerisinde özetlenmiştir.

Tablo 2.7. Federe öğrenmenin avantajları ve dezavantajları.

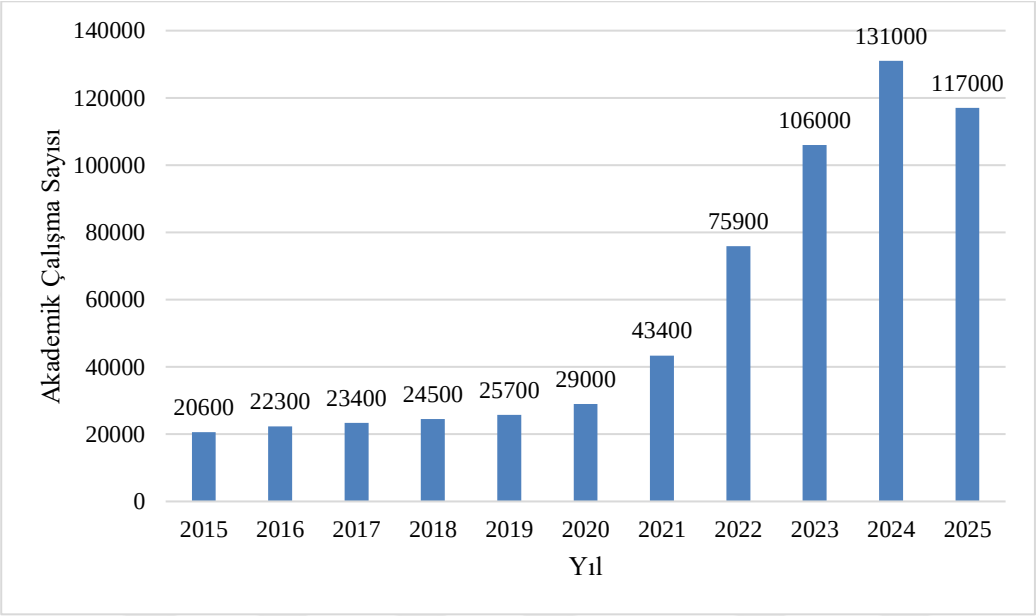
Avantajlar	Zorluklar ve Dezavantajlar
Veri Mahremiyeti: Ham veriler cihaz dışına çıkarılmadığı için veri güvenliği ve kullanıcı mahremiyeti güvence altına alınır.	**İstatistiksel Heterojenlik**: İstemcilerdeki verilerin bağımsız ve özdeş dağılmamış olması (Non-IID), modelin başarımını ve yakınsamasını olumsuz etkiler.
Düşük İletişim Maliyeti: Yalnızca model parametreleri iletildiği için ağ üzerindeki veri trafiği büyük oranda azalır.	**Sistemsel Heterojenlik:** İstemcilerin donanım kapasitelerindeki (işlemci, bellek, batarya vb.) farklılıklar, eğitim sürelerinde asenkronizasyona neden olur.
Enerji ve Hız Verimliliği: Sunucuya büyük boyutlarda veri yüklenmemesi, iletişim sürelerini kısaltır ve iletişim tabanlı enerji tüketimini düşürür.	**Karmaşık Optimizasyon:** Geleneksel modellere kıyasla iletişim turu, güncelleme sıklığı ve durdurma kriterleri gibi parametrelerin ayarlanması çok daha karmaşıktır.

FL model eğitiminin doğrudan verinin üretildiği istemcilerde gerçekleştirilmesiyle veri mahremiyetini yapısal olarak güvence altına almaktadır. Merkezi sunucuya istemci verisi yerine yalnızca yerel model parametrelerinin iletilmesi ağ trafiğini ve iletişim maliyetlerini önemli ölçüde düşürürken, aynı zamanda düşük enerji tüketimi ile hızlı iletişime olanak tanımaktadır.

Kavramsal olarak basit bir çalışma prensibine dayanmasına rağmen, FL sistemlerinin pratik uygulamaları çeşitli istatistiksel ve donanımsal kısıtlar içermektedir. Özellikle ağdaki istemciler arasındaki veri dağılımının heterojen (Non-IID) yapıda olması ve cihazların donanım kapasitelerindeki farklılıklar, global modelin eğitim performansını doğrudan ve kritik şekilde etkilemektedir. Bu dezavantajların üstesinden gelebilmek ve sistem kararlılığını sağlamak adına iletişim turu sayısı, eğitim sonlandırma kriterleri, yerel model güncelleme sıklığı ve global modelin mimari yapısı gibi sistem parametrelerinin, uygulamanın teknik gereksinimlerine göre titizlikle analiz edilip optimize edilmesi gerekmektedir.

FL alanında geliştirilen yöntem ve yaklaşımların sayısında yıllar içerisinde belirgin bir artış olduğu gözlemlenmektedir. Bu eğilimi görselleştirmek ve literatürün genel gelişim seyrini değerlendirmek amacıyla, konuya ilişkin akademik çalışmaların nicel bir incelemesi Şekil

2.12. ile sunulmuştur. İlgili grafik, 2015–2025 yılları arasında Google Scholar veri tabanında "Federated learning" anahtar kelimesi kullanılarak elde edilen sonuçların yıllara göre dağılımını göstermektedir. Bu dağılım, içerik uygunluğu açısından detaylı bir filtreleme yapılmaksızın oluşturulduğundan, kesin bir literatür analizi sunmaktan ziyade alanın genel araştırma eğilimini yansıtmayı amaçlamaktadır. Şekil incelendiğinde, federe öğrenmeye yönelik akademik ilginin yıllar içinde arttığı ve yayımlanan çalışma sayısında yükseliş eğilimi olduğu görülmektedir.



Şekil 2.12. "Federated learning" anahtar kelimesi kullanılarak Google Scholar veri tabanında yapılan arama sonuçlarının 2015–2025 yılları arasındaki yıllara göre dağılımı.

FL alanındaki çalışmaların daha sistematik bir şekilde incelenebilmesi amacıyla, literatürde yer alan araştırmalar alanlarına ve çalışma türlerine göre sınıflandırılarak Tablo 2.8. ile sunulmuştur. Hazırlanan bu tabloda her bir çalışmanın referans bilgisi, çalışma alanı ve çalışma kapsamına ilişkin açıklaması yer almaktadır. Açıklama sütununda, inceleme türündeki çalışmalar için hangi konuların analiz edildiği, araştırma çalışmalarında ise önerilen yöntemler, kullanılan veri setleri ve elde edilen performans sonuçları özetlenmiştir.

Tablo 2.8. Federe öğrenme çalışmalarının uygulama alanlarına göre sınıflandırılması.

Çalışma Alanı	Referans	Özet Açıklama
Sağlık	[155]	Blockchain tabanlı COVID-19 erken uyarı sistemi geliştirilmiştir.
	[156]	COVID-19 hastaları için klinik sonuç ve oksijen ihtiyacı tahmini yapılmıştır.
	[157]	Kanser teşhisi ve tedavi desteği için federated model

		önerilmiştir.
Öneri Sistemleri	[158]	Kullanıcı seviyesinde dağıtık matris ayrıştırma modeli geliştirilmiştir.
	[159]	Kampüs ortamı için kişiselleştirilmiş öneri sistemi geliştirilmiştir.
	[160]	FedAvg'i iyileştiren Hızlı Federe Öğrenme (Federated Fast Learning - FedFast algoritması önerilmiştir.
Akıllı Şehirler	[126]	Akıllı şehirlerde FL uygulanabilirliği ve açık problemler analiz edilmiştir.
	[161]	PM2.5 tahmini için FL ortamında istemciler arasında iletilen model güncellemelerini sıkıştırarak iletişim maliyetini azaltmayı ve sistem verimliliğini artırmayı amaçlayan bir yöntem geliştirilmiştir.
	[162]	Trafik akışı için uzamsal ve zamansal FL modeli geliştirilmiştir.
Finans ve Sigorta	[163]	VFL tabanlı gizlilik korumalı kredi riski değerlendirme modeli geliştirilmiştir.
Uç Bilişim	[164]	Uç cihazlar için FL çerçevesi önerilmiştir.
	[165]	Uyarlanabilir öğrenme oranı ile eğitim verimliliği artırılmıştır.
	[166]	Asenkron FL ile hesaplama ve iletişim maliyeti azaltılmıştır.
Saldırı Tespiti	[167]	Kurumlar arası veri paylaşmadan federe saldırı tespit yöntemi geliştirilmiştir.
	[168]	Uzun Kısa Süreli Bellek (Long Short Term Memory - LSTM) tabanlı federe saldırı tespit modeli önerilmiştir.
	[169]	Federe saldırı tespit mekanizması ile veri gizliliği korunmuştur.

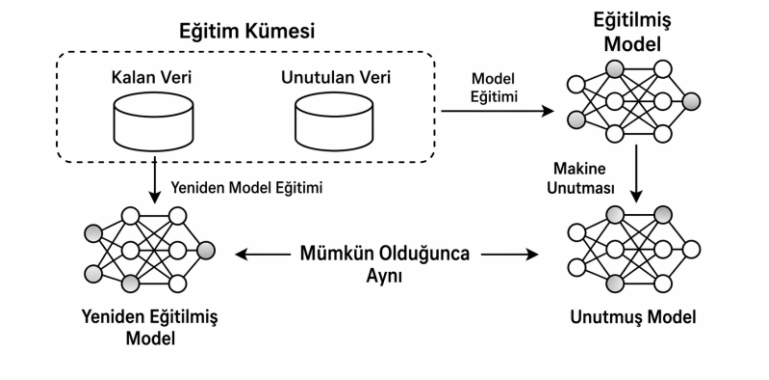
Federe öğrenme, veri gizliliğini koruyarak dağıtık ortamlarda model eğitimi yapılmasına olanak tanıyan etkili bir yaklaşım sunmaktadır. Ancak, bu yapıda kullanıcı verilerinin model üzerindeki etkisinin sonradan geri alınabilmesi önemli bir gereksinim olarak ortaya çıkmaktadır. Özellikle "unutulma hakkı" kapsamında, belirli verilerin modelden çıkarılması ihtiyacı, FL ortamında yeni araştırma problemlerini gündeme getirmiştir. Bu doğrultuda, bir sonraki bölümde FL mimarileri içerisinde geliştirilen federe unutma yaklaşımları ele alınacaktır.

2.3. Federe Öğrenme Ortamında Makine Unutması (Federated Unlearning)

Federe Unutma (Federated Unlearning - FU), FL sistemleri içerisindeki veri silme zorluklarıyla başa çıkmak amacıyla MU kavramından evrilerek ortaya çıkmış bir yöntemdir. Temel işlevi, FL modelinin belirli bir istemciyi veya o istemciye ait tanımlanabilir bilgileri kendi üzerinden unutmasını sağlamaktır.

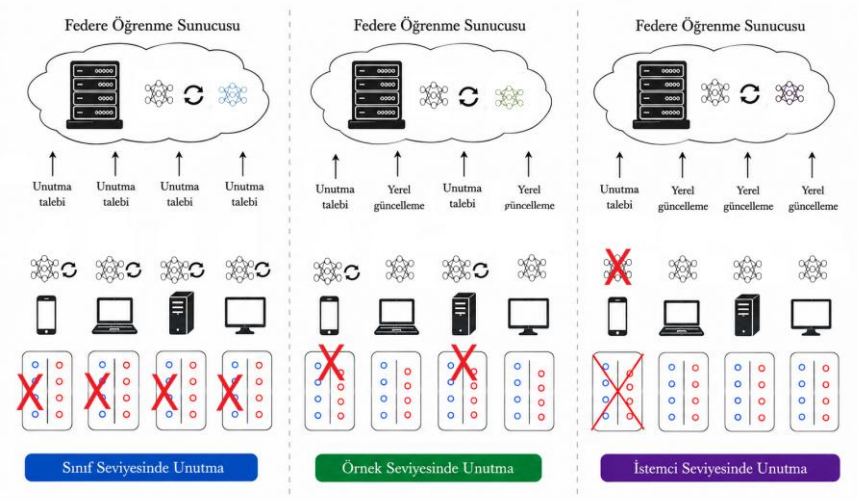
Şekil 2.13.'te MU ve yeniden eğitim süreçlerinin şematik gösterimi yer almaktadır. Şekilde yer alan silinecek veriler çıkarıldıktan sonra eğitilmiş modeli tamamen bir kenara bırakıp kalan verilerle modeli sıfırdan yeniden eğitmek işlemi, hesaplama açısından oldukça maliyetlidir. Bunun aksine, mevcut eğitilmiş model üzerinden bir unutma süreci yürüterek

işlemlere devam eden unutma yaklaşımları çok daha uygun maliyetli bir çözüm sunmaktadır. Bu unutma sürecinin temel amacı ise, unutma işlemi uygulanmış modelin, sıfırdan yeniden eğitilmiş bir modelle eşdeğer bir performans seviyesine ulaşmasını garanti altına almaktır.



Şekil 2.13. Makine unutması ve yeniden eğitim süreçlerinin şematik gösterimi.

Merkezi MU yaklaşımlarından farklı olarak FU sistemleri verisi silinecek olan hedef istemci, geriye kalan diğer istemciler ve merkezi sunucu olmak üzere birden fazla aktörü barındırır. Unutma algoritması, sürecin mimari tasarımına bağlı olarak bu taraflardan herhangi biri tarafından yürütülebilir. Sisteme iletilen unutma talepleri, hedeflenen verinin kapsamına göre üç temel düzeyde ele alınmaktadır. Belirli bir sınıfa ait tüm verilerin modelden dışlanmasını ifade eden sınıf seviyesi, spesifik bir istemcinin tüm yerel veri setinin kaldırılmasını içeren istemci seviyesi ve yalnızca belirli veri noktalarının etkisinin yok edilmesini amaçlayan örnek seviyesidir. Şekil 2.14.' te federe unutma için sınıf düzeyinde, örnek düzeyinde ve istemci düzeyinde unutma senaryoları iş akışları anlatılmaktadır [17].

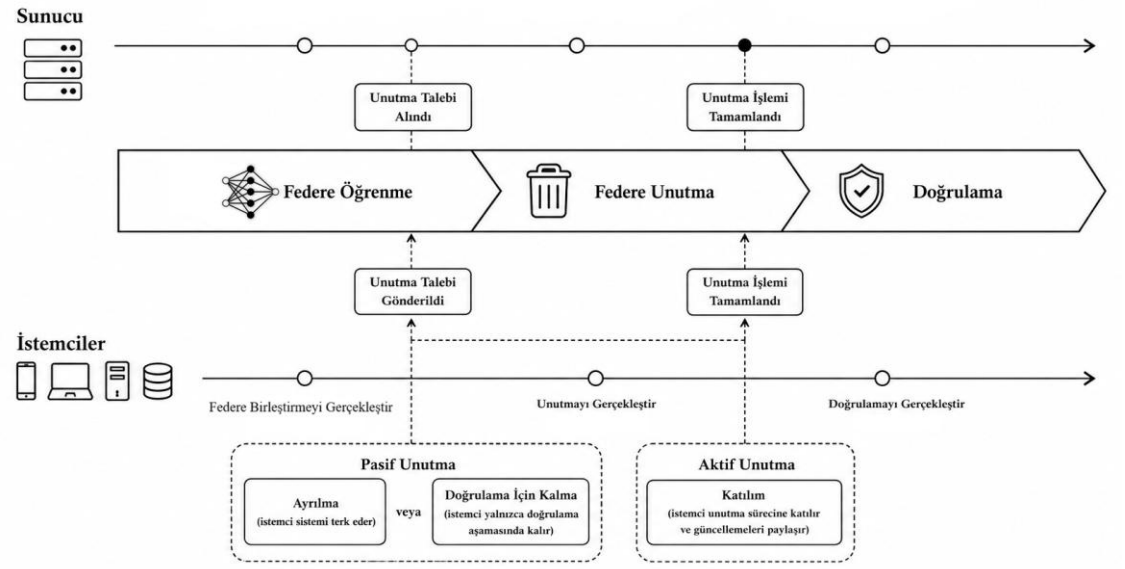


Şekil 2.14. Federe unutmanın; sınıf düzeyinde, örnek düzeyinde ve istemci düzeyinde iş akışları [17].

Temel bir FU çerçevesi, ham verilerin cihazlardan fiziksel olarak silinmesinden ziyade, bu verilerin eğitilmiş global model üzerinde bıraktığı istatistiksel ve kalıntı etkinin ortadan kaldırılmasına odaklanır. Bu doğrultuda süreç hedeflenen verinin yerel güncellemelerle modelden dışlanması, global modelin birleştirme yoluyla yeniden düzenlenmesi ve son olarak işlemin başarıyla gerçekleştiğinin doğrulanması adımlarından oluşan kapsamlı bir iş akışına dayanır. Temel amaç, modelin genel performansını korurken unutulan verinin izlerini sistemden tamamen ve güvenli bir şekilde arındırmaktır [16].

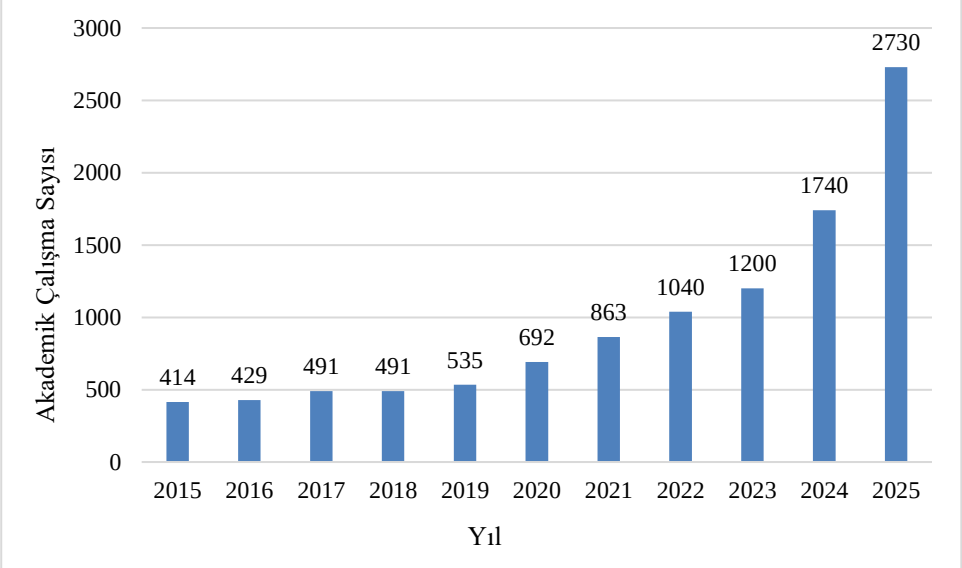
Şekil 2.15. FU sürecini göstermektedir. Bu iş akışı, modelin öğrenmesini, verinin unutulmasını ve doğrulama aşamalarını kronolojik sırayla açıklamaktadır. FU sistemi, unutma talebi aldığında mimari yapıya bağlı pasif unutma ve aktif unutma olarak iki temel stratejiden birini izler. Pasif unutma, hedef istemcinin talep sonrasında sistemden ayrıldığı ve unutma sürecinin diğer bileşenler tarafından yürütüldüğü yaklaşımdır. Pasif unutma için sunucu tarafından gerçekleştirilen unutma senaryosunda, gradyanlar ve global model gibi geçmiş bilgiler saklanır ve sunucu, çeşitli yöntemler kullanarak unutulan istemcinin etkisini modelden çıkarır. İstemci destekli unutma senaryosunda ise, standart FL iş akışı izlenir ve kalan istemcilerden gelen güncellenmiş bilgiler birleştirilerek global model yinelemeli olarak iyileştirilir. Aktif unutma**,** hedef istemcinin unutma sürecine aktif olarak katıldığı ve sürecin ardından doğrulama yaparak ya da yapmadan sistemde kalma veya ayrılma seçeneğine sahip olduğu durumu ifade etmektedir. Unutma talepleri, hedeflenen amaca bağlı olarak istemci veya merkezi sunucu tarafından başlatılabilir. Unutma ve doğrulama süreçleri, sistem tasarımına göre merkezi sunucu, hedef istemciler, diğer istemciler veya bu bileşenlerin birlikte çalışmasıyla gerçekleştirilebilir [16].

MU çalışmalarında, önerilen yöntemlerin ne ölçüde etkili olduğunu belirlemek amacıyla çeşitli doğrulama metrikleri kullanılmaktadır. FU çalışmalarında doğrulama metrikleri; doğruluk temelli ölçütler, çalışma süresi analizleri, arka kapı saldırısı (backdoor attack) ve üyelik çıkarım saldırısı (membership inference attack) gibi saldırı tabanlı değerlendirmeler ve unutmuş model ile yeniden eğitilmiş model arasındaki farkın incelenmesini kapsamaktadır. Ancak literatürde unutmanın kanıtlanmasına yönelik standart ve yaygın kabul görmüş yöntemler bulunmamaktadır [16].



Şekil 2.15. Federe unutma iş akış şeması [16].

FU alanında literatürdeki eğilimi ortaya koymak amacıyla, 2015–2025 yılları arasında bu alanda yayımlanan çalışmaların yıllara göre dağılımı Şekil 2.16. ile gösterilmektedir. Bu grafik, yalnızca "Federated unlearning" anahtar kelimesi ile yapılan genel arama sonuçlarına dayanmaktadır ve çalışmaların içerik uygunluğu detaylı olarak filtrelenmemiştir. Bu nedenle, elde edilen dağılım alanın genel araştırma eğilimini göstermek amacıyla sunulmaktadır. Şekil incelendiğinde, federe unutmaya yönelik akademik ilginin yıllar içinde arttığı ve yayımlanan çalışma sayısında son yıllarda belirgin bir yükseliş olduğu görülmektedir.



Şekil 2.16. "Federated unlearning" anahtar kelimesi kullanılarak Google Scholar veri tabanında yapılan arama sonuçlarının 2015–2025 yılları arasındaki yıllara göre dağılımı.

Tablo 2.9. içerisinde literatürde yer alan FU yaklaşımları; unutma türü (aktif/pasif), unutma işlemini gerçekleştiren taraf, unutulan veri kapsamı ve doğrulama mekanizmaları açısından karşılaştırmalı olarak sunulmuştur. Tablo, farklı FU yöntemlerinin hangi senaryolarda tercih edildiğini ve yöntemler arasındaki temel farklılıkları ortaya koyarak literatürdeki mevcut yaklaşımlara bütüncül bir bakış sağlamayı amaçlamaktadır.

Tablo 2.9. FU yöntemlerinin karşılaştırmalı özeti.

Referans	Unutma Türü	Kim Unutur?	Ne Unutulur?	Doğrulayan	Doğrulama Yöntemleri
[170]	Pasif	Sunucu	Hedef istemci	Yok	Doğruluk, Unutma Süresi
[37]	Pasif	Sunucu İstemciler	Hedef istemci	Yok	Tur sayısı, Doğruluk
[171]	Pasif	Sunucu	Hedef istemci	İstemci	Doğruluk
[84]	Aktif	Sunucu İstemciler	Kısmi veri	Sunucu	Tahmin sonuçları
[172]	Aktif	Tüm İstemciler	Veri / Sınıf	Yok	Doğruluk, Unutma Süresi
[173]	Aktif	Hedef istemci	Kısmi veri	Yok	Doğruluk, Unutma Süresi

Tablo 2.9. ile sunulan karşılaştırmanın ardından, literatürde öne çıkan FU yöntemleri aşağıda detaylı olarak ele alınmaktadır.

Zehirleme Saldırıları ve İhlallerden Sertifikalı Kurtarma (Certified Recovery from Poisoning Attacks and Breaches - Crab) yöntemi, zararlı istemcilerin tespit edilmesinden sonra modeli baştan eğitmek yerine, seçili geçmiş bilgileri kullanarak ve uygun bir önceki modele geri dönerek global modeli hızlı bir şekilde yeniden oluşturmayı amaçlar. Bu yaklaşım, tüm geçmiş veriyi kullanma ihtiyacını ortadan kaldırarak hesaplama maliyetini ve bellek kullanımını azaltırken**,** model performansını korur [170].

Sıralı Bilgilendirilmiş Federe Unutma (Sequential Informed Federated Unlearning -SIFU) yöntemi, federe öğrenmede bir istemcinin katkısını modelden çıkarmak için, FedAvg ile uyumlu çalışan ve hem konveks hem de konveks olmayan optimizasyon durumlarında uygulanabilen bir yaklaşımdır. Bu yöntem, istemciler için ek hesaplama maliyeti oluşturmadan unutma işlemini gerçekleştirirken, aynı zamanda unutmanın kalitesine ilişkin teorik garantiler sunmayı amaçlamaktadır [37].

Doğrulanabilir Federe Unutma (Verifiable Federated Unlearning – VeriFi) yöntemi, federe öğrenmede unutma işleminin yalnızca gerçekleştirilmesini değil, aynı zamanda doğrulanabilir olmasını hedefleyen bir çerçeve sunmaktadır. Bu yaklaşımda, sistemden ayrılan istemciye doğrulama hakkı tanınmakta ve özel olarak tasarlanmış işaretleyiciler üzerinden modelin davranışı analiz edilerek unutma etkisinin gerçekten gerçekleşip gerçekleşmediği kontrol edilmektedir [171].

Bilgi Grafikleri için Federe Öğrenme Tabanlı Unutma (Federated Learning-based Unlearning for Knowledge Graphs – FedLU) yöntemi, bilgi grafı temsilleri için FL ortamında hem öğrenme hem de unutma süreçlerini birlikte ele alan bir çerçeve sunmaktadır. Bu yaklaşım, veri heterojenliğini azaltmak için karşılıklı bilgi damıtma kullanırken, unutma sürecinde geriye dönük müdahale ve pasif unutma mekanizmalarını birleştirerek belirli bilgilerin modelden etkili şekilde silinmesini sağlamayı amaçlamaktadır [84].

Veri Damıtma Tabanlı Federe Unutma (Dataset Distillation-based Federated Unlearning – QuickDrop) yöntemi, federe öğrenmede unutma sürecini hızlandırmak için veri damıtma (dataset distillation) kullanan bir yaklaşımdır. Bu yöntemde istemciler, orijinal veriyi temsil eden küçük veri setleri oluşturur ve unutma işlemini bu kompakt veri üzerinden gradyan yükseltme ile gerçekleştirerek hesaplama maliyetini önemli ölçüde azaltmayı amaçlamaktadır [172].

Bayesçi Federe Unutma (Bayesian Federated Unlearning – BFU) yöntemi, FL ortamında ham veriyi paylaşmadan unutma işlemini gerçekleştirmek için Bayesçi bir yaklaşım sunmaktadır. Bu yöntem, unutma oranı ile unutma ve model bilgisini koruma arasındaki dengeyi ayarlarken, doğruluk kaybını azaltmak amacıyla unutma ve öğrenme süreçlerini birlikte optimize etmeyi hedeflemektedir [173].

Tez çalışmasında kullanılan, MU ile FL yaklaşımlarının birleşimini temsil eden federe unutma kavramı, teorik temelleri ve literatürde önerilen çalışmalar çerçevesinde ele

alınmıştır. Bu kapsamda, mevcut yöntemlerin temel prensipleri, uygulama alanları ve değerlendirme yaklaşımları incelenerek alanın genel çerçevesi ortaya konulmuştur. Ayrıca, literatürdeki çalışmalar doğrultusunda federe unutma yaklaşımlarının gelişim süreci ve araştırma eğilimleri değerlendirilmiştir.

2.4. İlişkisel Sınıflandırma (Associative Classification)

Makine öğrenmesi, veri setlerinden anlamlı örüntüler çıkararak veri analizi, tahminleme ve karar destek sistemlerinde yaygın olarak kullanılan temel bir yaklaşımdır [174]. Geleneksel sınıflandırma yöntemleri, yapılandırılmış veri setleri üzerinde yüksek doğruluk oranları sunarak literatürde büyük başarılara imza atmıştır. Bu standart algoritmalar uzun yıllar boyunca sınıflandırma problemlerinde çok güçlü bir temel oluşturmuştur. Buna karşın, geleneksel modellerin en temel kısıtlarından biri, veri örnekleri arasındaki olası bağımlılıkları göz ardı ederek her bir girdiyi bağımsız olarak ele almalarıdır. Oysa gerçek dünya problemlerinin birçoğu, verilerin yalnızca kendi içsel öznitelikleriyle değil, diğer verilerle aralarındaki bağlarla da anlam kazandığı karmaşık ağ yapıları barındırır. İlişkisel sınıflandırma [175] teknikleri, veriler arasındaki bu bağlantıları ve bağlamsal ilişkileri modelleme sürecine doğrudan dahil ederek geleneksel yöntemlerin tıkandığı bu sorunu çözer. İlişkisel veri üzerinden öğrenen bu yaklaşım, hedefe yönelik daha doğru ve anlamlı sonuçlar elde edilmesini sağlar. Özellikle sağlık, eğitim ve finans gibi alanlar, verilerindeki yoğun içsel bağımlılıklar nedeniyle ilişkisel modellemeden büyük ölçüde faydalanmaktadır.

İlişkisel sınıflandırma temelde, sınıflandırma yöntemi ile birliktelik kuralları yöntemini birleştiren bir veri madenciliği yaklaşımıdır. Bu entegre mimarinin daha iyi anlaşılabilmesi için öncelikle sınıflandırma ve birliktelik kuralı kavramlarının teorik altyapılarının incelenmesi gerekmektedir.

Sınıflandırma, daha önceden etiketlenmiş geçmiş verilerden elde edilen tecrübe ile sınıfı bilinmeyen yeni verilerin kategorisini/sınıf etiketini tahmin etme işlemidir [176]. Makine öğrenmesi algoritması bu ilişkiyi öğrendiğinde, yeni gelen verilerin hangi kategoriye ait olduğuna karar verir. Bu öğrenme süreci, ulaşılmak istenen hedefin niteliğine bağlı olarak iki farklı şekilde ele alınır. Sonucun sadece iki seçenekten oluştuğu durumlara ikili sınıflandırma (binary classification), ikiden fazla kategorinin bulunduğu durumlara ise çok sınıflı sınıflandırma (multiclass classification) adı verilmektedir [177] .

Sınıflandırma modeli oluşturmak için veri seti eğitim ve test veri seti olarak iki ana parçaya ayrılmaktadır. Veri seti içerisinde modele girdi sağlayan öznitelikler ve bu özniteliklere ait

sonuç bilgisi yer alır. Eğitim verisi üzerine algoritmik bir yöntem uygulanarak sistemin veri içindeki örüntüleri öğrenmesi sağlanır ve model inşa edilir. Elde edilen bu modelin performansı test veri seti üzerinden ölçülür. Aşırı öğrenme probleminden arındırılmış ve başarılı bulunan model, daha sonra sınıfı bilinmeyen yeni veriler üzerinde tahmin işlemini gerçekleştirerek nihai kategorik çıktıları üretir.

Literatürde makine öğrenmesi tabanlı sınıflandırma yapan yöntemler, algoritmik yaklaşımlarına göre genellikle şu temel kategorilere ayrılmaktadır [178]:

- **İstatistiksel ve Olasılıksal Yöntemler:** Lojistik Regresyon, Naive Bayes Sınıflandırıcısı, Lineer Regresyon.
- **Mesafeye ve Geometriye Dayalı Yöntemler:** K-En Yakın Komşu Algoritması, Destek Vektör Makineleri.
- **Mantıksal ve Ağaç Tabanlı Yöntemler:** Karar Ağaçları, Rastgele Orman, Gradyan Arttırma Makineleri.
- **Sinir Ağları Tabanlı Yöntemler:** Çok Katmanlı Algılayıcılar, Yapay Sinir Ağları.
- **Kural Tabanlı Yöntemler:** Sıfır Kuralı (Zero Rule), Tek Kural / Tek Özellik Kuralı (One Rule), Karar Listeleri.

Birliktelik kuralları analizi, veri madenciliğinde kullanılan tekniklerden birisidir [179]. Birliktelik kuralı, veri seti içerisinde birbiriyle ilişkili olan özelliklerin veya öğe kümelerinin eş zamanlı ortaya çıkma davranışlarını tespit eden bir yöntemdir. Birliktelik davranışlarının keşfedilmesi, geleceğe yönelik stratejik çıkarımlar yapılmasını sağlamaktadır.

Bu yöntemin literatürdeki en tipik örneği market sepeti analizi uygulamasıdır [180]. Bu uygulamada amaç, alışveriş esnasında müşterilerin sepetlerindeki ürünler arasındaki birliktelik ilişkisini bulmak ve satın alma alışkanlıklarını modellemektir. Keşfedilen bu gizli bağıntılar sayesinde satıcılar çapraz satış ve pazarlama avantajları elde ederler.

Yöntem uygulanırken veri setinden matematiksel olarak $X \rightarrow Y$ formunda ifade edilen kurallar oluşturulur. Bu kural: *"Eğer X var ise Y' de vardır."* şeklinde okunur. Burada sol tarafta bulunan X kuralın öncülünü, sağ tarafta bulunan Y ise kuralın sonucunu temsil etmektedir. Üretilen aday kuralları değerlendirebilmek, filtrelemek ve sınıflandırma gücünü ölçebilmek için literatürde başlıca şu istatistiksel metrikler kullanılmaktadır [181]:

- **Destek (Support):** İlgili öğe kümesinin (X ve Y'nin birlikte, $X \rightarrow Y$) tüm veri seti içerisinde ne sıklıkla gerçekleştiğini ifade eder. Bir kuralın veri setindeki temsiliyet gücünü

ve evrenselliğini gösterir. Düşük destekli kurallar genellikle tesadüfi veya çok nadir durumlar olarak kabul edilir. Denklem (1)'de destek formülü yer almaktadır.

$$Destek = \frac{Frekans(X,\ Y)}{N} \qquad (1)$$

- **Güven (Confidence):** *X*'in gerçekleştiği durumlarda *Y*'nin de gerçekleşme olasılığını gösterir. Kuralın ne kadar güvenilir ve tutarlı olduğunun temel ölçüsüdür. Denklem (2)'de destek formülü yer almaktadır.

$$Güven = \frac{Frekans(X,\ Y)}{Frekans(X)} \qquad (2)$$

Denklem (1) ve Denklem (2)'de *X* ve *Y* sırasıyla değerlendirilen kuralın öncül ve sonuç bileşenlerini, $X \rightarrow Y$'yi temsil etmektedir. *N*, istemcinin veri setindeki toplam örnek sayısını ifade etmektedir. *Frekans (X, Y)*, *X* ve *Y*'nin birlikte görülme sıklığını ve *Frekans(X)* ise *X*'in tek başına görülme sıklığını ifade etmektedir.

- **Kaldıraç (Lift):** *X* ve *Y*'nin birbirinden istatistiksel olarak bağımsız olduğu varsayımına kıyasla, birlikte görülme olasılıklarının ne kadar arttığını ölçer. Lift değerinin 1'e eşit olması değişkenler arasında bir ilişki olmadığını, 1'den büyük olması ise *X* ile *Y* arasında rastlantısal olmayan, pozitif yönde bir korelasyon bulunduğunu göstermektedir. Denklem (3)'te destek formülü yer almaktadır.

$$Kaldıraç = \frac{Güven\ (X,\ Y)}{Destek(Y)} \qquad (3)$$

Veri tabanındaki örüntülerin keşfedilmesi süreci, genellikle uygulama tasarımcısı veya alan uzmanı tarafından önceden belirlenen minimum destek ve minimum güven eşik parametrelerine dayanmaktadır. Bu istatistiksel eşikleri sağlayan aday öğe kümeleri analiz edilerek modelin temelini oluşturacak anlamlı birliktelik kuralları elde edilir.

Literatürde bu madencilik sürecini gerçekleştirmek amacıyla, farklı arama stratejilerine ve bellek yönetimi mimarilerine sahip birçok algoritma geliştirilmiştir. Alan literatüründe en yaygın kullanılan, performansı ile öne çıkan ve modern yöntemlere referans kabul edilen üç temel algoritma şunlardır [180]:

Apriori: Aday üretimi mantığına dayanan ve veri tabanını yatay olarak tarayan en klasik yaklaşımdır.

Frequent Pattern Growth (FP-Growth): Apriori'nin en büyük dezavantajı olan yüksek maliyetli aday üretimi ve veri tabanını defalarca tarama zorunluluğunu ortadan

kaldıran algoritmadır. Veriyi özel bir ağaç yapısında sıkıştırarak bellek ve işlem süresi açısından çok daha yüksek bir verimlilik sunar.

Equivalence Class Transformation (Eclat): Veri tabanını geleneksel yatay format yerine dikey formatta kodlayarak işleyen algoritmadır. Derinlik öncelikli arama stratejisini ve küme kesişim işlemlerini kullanarak kural keşif sürecini ciddi ölçüde hızlandırır.

Tablo 2.10.'da öne çıkan bu üç temel algoritmanın karşılaştırması yer almaktadır.

Tablo 2.10. Apriori, Eclat ve FP-Growth algoritmalarının karşılaştırması.

Özellik	Apriori	Eclat	FP-Growth
Çalışma Prensibi	Aday kümeler üreterek yinelemeli bir şekilde yaygın öğe kümelerini bulur.	Dikey veri gösterimi ile yaygın öğeleri hızlıca bulur.	Frekans örüntü ağaç kullanarak veri setindeki yaygın öğeleri arar.
Veri Gösterimi	Yatay veri gösterimi kullanır.	Dikey veri gösterimi kullanır.	Veriler frekans örüntü ağacı yapısıyla yeniden yapılandırılır.
Performans	Büyük veri setlerinde yavaş olabilir, aşırı gereksiz işlem yapar.	Küçük veri setlerinde hızlıdır, büyük veri setlerinde bellek kullanımı sorun yaratabilir.	Büyük veri setlerinde en hızlısıdır.
Gereksiz Aday Küme Sayısı	Yüksek	Orta	Düşük
Bellek Kullanımı	Düşük	Orta	Yüksek
Uygulama Alanları	Küçük ve orta ölçekli veri setlerinde kullanıma uygundur.	Küçük veri setlerinde hızlı analiz sağlar.	Büyük veri setlerinde yüksek performans sunar.
Kurulum / Uygulama Karmaşıklığı	Düşük	Orta	Yüksek
Avantajları	Basit ve anlaşılırdır, uygulanması kolaydır.	Dikey veri yapısı sayesinde bazı durumlarda daha hızlı sonuçlar sunar.	Performans ve hız açısından en etkili yöntemdir.
Dezavantajları		Büyük veri setlerinde bellek	

	Aşırı gereksiz işlem yapar, büyük veri setlerinde yavaştır.	kullanımı zorlayıcı olabilir.	Yapısı karmaşıktır, daha fazla bellek tüketir.

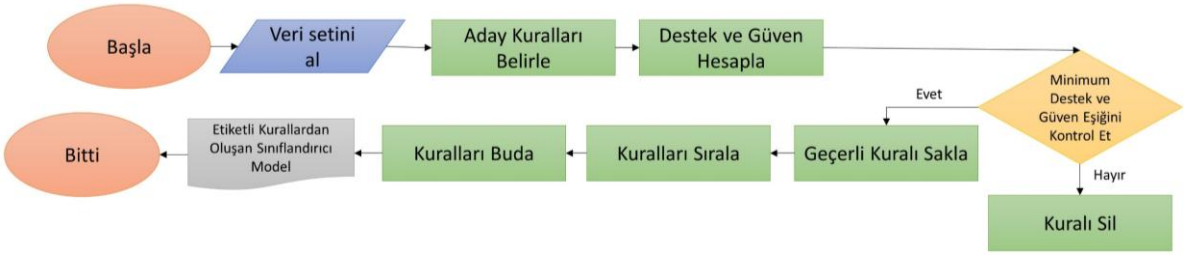
Bu temel mimarilerin yanı sıra literatürde; büyük veri setleri için dağıtık, paralel veya farklı veri işleme gereksinimlerine özel olarak geliştirilmiş Off-line Candidate Determination (OCD), Partitioning, Sampling, Algorithm Continuous Association Rule Mining Algorithm (CARMA), Apriori-TID, Apriori-Hybrid ve Recursive Elimination gibi spesifik algoritmalar da bulunmaktadır [182], [183].

İlişkisel sınıflandırma, yukarıda bahsedilen sınıflandırma ve birliktelik kuralları tekniklerini, Sınıf Birliktelik Kuralları (Class Association Rules - CAR) formunda bir araya getirir [175]. Bir CAR, genel olarak $X \rightarrow Y$ şeklinde yazılır. Standart birliktelik kurallarından farklı olarak burada kuralın sağ tarafı (Y) rastgele bir öğe değil, doğrudan tahmin edilmek istenen sınıf etiketidir.

Şekil 2.17.'de ilişkisel sınıflandırma diyagramı yer almaktadır. Şekle göre ilişkisel sınıflandırma algoritmaları yapısal olarak iki ana aşamadan oluşmaktadır:

Kural Üretimi (Rule Generation): Eğitim veri setindeki minimum destek ve minimum güven eşiklerini sağlayan tüm olası sınıf birliktelik kurallarının keşfedildiği aşamadır. Bu aşamada çoğunlukla Apriori mantığı kullanılarak hedef sınıfı işaret eden sık görülen öğe kümeleri üretilir.

Sınıflandırıcı İnşası (Classifier Building): Üretilen kurallar içerisinden, sınıflandırma performansına en yüksek etkiyi yapacak olanların seçilip gereksiz olanların budandığı ve nihai karar listesinin oluşturulduğu aşamadır.



Şekil 2.17. İlişkisel sınıflandırma yöntemi akış diyagramı.

İlişkisel sınıflandırmada kullanılan birçok algoritma bulunmaktadır [20]. Bu amaçla geliştirilen algoritmaların temelini Birliktelik Kurallarına Dayalı Sınıflandırma (Classification Based on Association - CBA) oluşturmaktadır. İlişkisel sınıflandırma,

birliktelik kurallarına dayalı olarak sınıflandırma yapan bir veri madenciliği yöntemidir ve elde edilen kurallar doğal yapısı gereği kullanıcı tarafından yorumlanabilmektedir.

Zaman içerisinde modelin doğruluk değerini, eğitim süresini ve farklı veri yapılarına uyum sağlama yeteneğini artırmak amacıyla CBA algoritması üzerinden çeşitli iyileştirmeler yapılmış ve literatüre yeni algoritmalar kazandırılmıştır [20]. Özellikle sürekli değişen talepler neticesinde değişime uğrayan veri kümelerinde, modelin sıfırdan yeniden eğitilmesini önlemek amacıyla artırımlı öğrenme [184] yöntemleri geliştirilmiştir.

Literatürde öne çıkan başlıca ilişkisel sınıflandırma algoritmaları çalışma prensiplerine göre aşağıda listelenmiştir:

Birliktelik Kurallarına Dayalı Sınıflandırma (Classification Based on Association - CBA): Literatürdeki ilk ilişkisel sınıflandırma algoritmalarındandır. Algoritmanın çalışma mantığı; verilerle kuralların oluşturulması, zayıf kuralların budanması ve en iyi sınıflandırma yapan kuralların elde edilerek karar listesine eklenmesi olmak üzere üç temel aşamadan oluşur.

Çoklu Sınıf Birliktelik Kurallarına Dayalı Sınıflandırma (Classification Based on Multiple Class Association Rules - CMAR): CBA'dan farklı olarak, tek bir kural yerine birden fazla kuralı eş zamanlı değerlendirerek esnek ve doğru bir sınıflandırma gerçekleştirir. Kural bağımsızlığını dikkate alır ve kararlarını güven ağırlıklı çoğunluk oylaması mekanizması ile verir [185].

Birliktelik Kuralına Dayalı Çok Sınıflı Sınıflandırma (Multiclass Classification Based on Association Rule - MCAR): CMAR ile benzer mantıkta çalışan ancak kural oluşturma süreçleri farklılık gösteren bir algoritmadır. Çok sınıflı problemleri çözmeyi hedefler ve geleneksel karar ağaçlarından daha yüksek sınıflandırma başarısı sunar [186].

Tahmin Edici Birliktelik Kurallarına Dayalı Sınıflandırma (Classification based on Predictive Association Rules - CPAR): CBA ve CMAR algoritmalarında karşılaşılan aşırı kural üretimi ve aşırı öğrenme problemlerini çözmek amacıyla geliştirilmiştir. Her bir veri örneği için yüksek güven değerine sahip birden fazla kuralı aynı anda kullanarak esnek bir tahmin yapısı sunar [20].

Birliktelik Kurallarına Dayalı Hızlı Sınıflandırma (Fast Classification Based on Associations - FCBA): Kural oluşturma sürecini hızlandırmak için dahili budama

tekniklerini kullanan ve bulanık mantık temellerine dayanan bir algoritmadır. Kesin eşik değerleri yerine bulanık kümeler kullanarak belirsizlik içeren verilerde daha doğru bir yapı sunar [187].

Ağırlıklı Birliktelik Kurallarına Dayalı Sınıflandırma (Weighted Classification Based on Associations - WCBA): Birliktelik kurallarına belirli bir önem derecesine göre ağırlık) atayan ve sınıflandırma sürecini bu ağırlıklı puanlar üzerinden gerçekleştiren algoritmadır. Öznitelik ağırlıklandırması ve harmonik ortalama gibi istatistiksel metrikleri de sürece dahil eder [188].

Çoklu Destekli CBA (Multiple Support CBA - msCBA): Kural seçimi ve sıralaması aşamalarında daha gelişmiş stratejiler kullanarak CBA algoritmasının doğruluğunu ve kararlılığını artırmayı hedefleyen algoritmadır. Özellikle büyük veri setlerinde daha dengeli sonuçlar üretir [20].

Artırımlı Madenciliğe Dayalı Birliktelik Tabanlı Sınıflandırma (Associative Classifier with Instance Matching - ACIM): Modelin sürekli güncel kalmasını sağlayan artırımlı öğrenme yöntemine dayanan bir algoritmadır. Yeni veri eklendiğinde sınıflandırıcıyı en baştan eğitmek yerine, örnek benzerliğine dayalı olarak kural setlerini hızlıca günceller [189].

Artırımlı Madencilik Algoritmasına Dayalı Geliştirilmiş Birliktelik Tabanlı Sınıflandırma (Enhanced Associative Classifier with Instance Matching - E-ACIM): ACIM algoritmasının iyileştirilmiş bir versiyonudur. Örnek benzerlik eşleşmelerini daha verimli hesaplayarak, doğruluk oranı ile çalışma süresi arasında daha iyi bir denge sunar [190].

Hibrit Birliktelik Tabanlı Sınıflandırıcı ile Sınıflandırma (Classification using Hybrid Association Classifier - CHAC): Kural sıralama, örnek eşleştirme ve güven analizi gibi farklı ilişkisel sınıflandırma stratejilerini bir araya getiren hibrit bir yaklaşımdır. Farklı veri tiplerine karşı çok daha uyumlu bir sınıflandırma modeli sunmayı hedefler [20].

Literatürde sunulan ve yukarıda detaylandırılan algoritmalar, ilişkisel sınıflandırma yaklaşımının farklı problem tiplerinde ve veri yapılarında kullanılabildiği gösterilmektedir. Bu tez çalışması kapsamında, ilişkisel sınıflandırmanın literatürdeki temelini oluşturan ve en yaygın kabul gören CBA algoritması temel alınmıştır. Diğer algoritmalar ise tezin devamı niteliğindeki gelecek çalışmalarda ele alınacaktır. Yapılacak çalışmada, FL

mimarisi üzerinde geliştirilen modele CBA yerine diğer algoritmaların entegre edilmesi planlanmaktadır. Böylece, farklı algoritmaların model performansına ve MU süreçlerine olan etkileri karşılaştırmalı olarak incelenebilecektir.

CBA algoritmasının güncel literatürdeki geçerliliğini ve kullanım yaygınlığını analiz etmek amacıyla spesifik bir literatür taraması gerçekleştirilmiştir. Algoritmanın modern araştırma problemlerindeki uygulanabilirliğini ve güncelliğini koruyup korumadığını tespit etmek hedeflenmiştir. Bu amaç doğrultusunda, Google Scholar veri tabanı üzerinden yalnızca 2025 yılında yayımlanmış akademik çalışmalar taranmıştır. Tarama sürecinde "Classification Based on Association" ve "CBA" anahtar kelimeleri kullanılmış olup Türkçe ve İngilizce dillerinde yayımlanan araştırma makaleleri, derlemeler ve konferans bildirileri dahil edilmiştir. Tarama metodolojisi iki aşamalı olarak yürütülmüştür. İlk aşamada elde edilen yayınlar özet uygunluğuna göre titizlikle filtrelenmiş, ikinci aşamada ise seçilen yayınların tam metin içerikleri incelenerek doğrudan CBA algoritmasını temel alan, geliştiren veya farklı bir disipline entegre eden çalışmalar tespit edilmiştir.

CBA algoritmasının literatürde hala aktif olarak kullanıldığını ve yeni araştırmalara temel oluşturduğunu kanıtlayan bu metodolojik tarama sonucunda elde edilen güncel çalışmalar, Tablo 2.11.'de özetlenmiştir.

Tablo 2.11. 2025 yılında CBA algoritması ve ilişkisel sınıflandırma üzerine gerçekleştirilen güncel çalışmalar.

Makale	Makale Özeti
[20]	İlişkisel sınıflandırma algoritmalarının literatürdeki performansını ölçmek amacıyla, aralarında CBA'nın da bulunduğu 9 farklı algoritmayı 4 standart veri seti üzerinde kıyaslayan kapsamlı bir benchmark çalışmasıdır.
[191]	Çalışmada, farklı sınıflandırma ve öznitelik seçimi yaklaşımları bir arada değerlendirilmiş ve çeşitli modellerin performansları karşılaştırılmıştır. Elde edilen yüksek doğruluk, kesinlik ve duyarlılık değerleri, önerilen yöntemlerin etkinliğini göstermektedir. Bu yönüyle çalışma, farklı yöntemlerin performansını ortaya koyan bir benchmark niteliği taşımaktadır.
[192]	Kanser alt tiplerini analiz etmek ve tıbbi biyobelirteçleri belirlemek amacıyla, çoklu omik verilerini ve ilişkisel çizge sinir ağlarını bir arada kullanan yeni bir sınıflandırma çerçevesi önermektedir.
[117]	FL mimarisinde veri gizliliğini sağlamak ve dağıtık ağlardaki çıkarım saldırılarını hafifletmek amacıyla rastgele yanıt mekanizmalarına dayalı

	diferansiyel mahremiyet yaklaşımlarını inceleyen bir yöntem çalışmasıdır.
[193]	Tıbbi karar destek sistemlerinde güvenilir ve hekimler tarafından yorumlanabilir kararlar sunmak amacıyla, kural çıkarımına dayalı açıklanabilir bir klinik karar çerçevesi geliştiren yöntem çalışmasıdır.
[194]	Hassas sağlık verilerinin gizliliğini korumak amacıyla merkezi veri toplama işlemi yerine, FL mimarisi üzerinden dağıtık gerçek dünya verileriyle model eğitimi ve analizi gerçekleştiren bir uygulama çalışmasıdır.

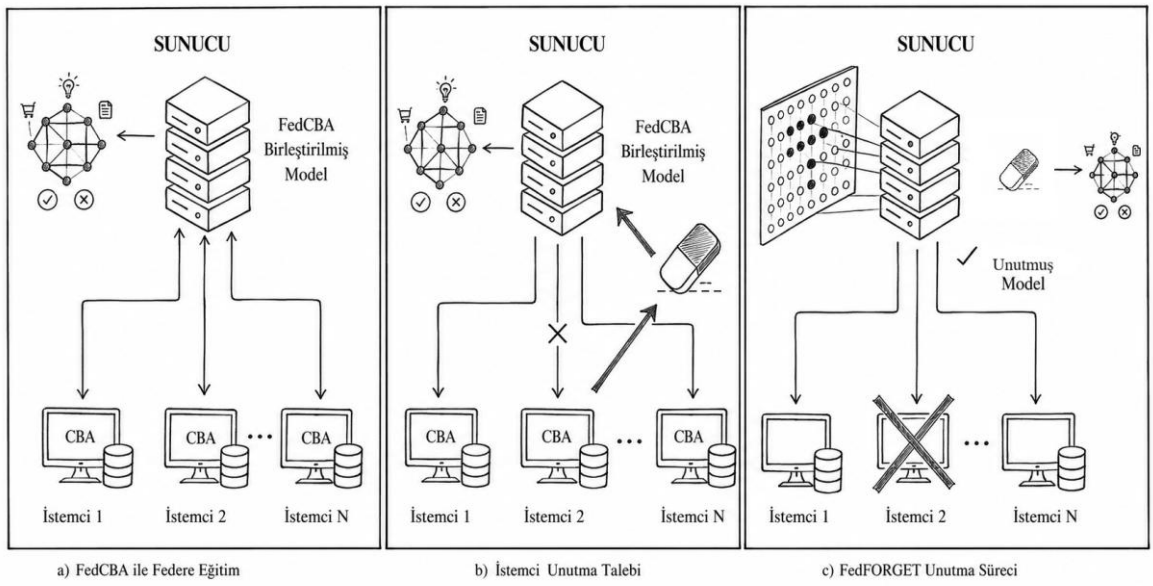
Bu bölümde, ilişkisel sınıflandırma yaklaşımı teorik temelleri ile ele alınmış ve ardından literatürde gerçekleştirilen güncel çalışmalar incelenmiştir. İncelenen çalışmalar, hem klasik ilişkisel sınıflandırma algoritmalarının performansını karşılaştıran benchmark analizlerini hem de farklı veri türleri ve uygulama alanlarında geliştirilen yöntemleri kapsamaktadır. Özellikle, yüksek doğruluk değerleri elde eden karşılaştırmalı çalışmalar ile birlikte, tıbbi veri analizi, klinik karar destek sistemleri ve gizlilik odaklı FL yaklaşımları gibi farklı bağlamlarda bu yöntemlerin kullanımına yer verildiği görülmektedir. Bu durum, ilişkisel sınıflandırma yaklaşımının yalnızca performans açısından değil, aynı zamanda yorumlanabilirlik ve uygulama çeşitliliği açısından da literatürde önemli bir yer edindiğini göstermektedir.

3. YÖNTEM

3.1. Sisteme Genel Bakış

Bu çalışma kapsamında, FL mimarisinde yeniden eğitime gerek kalmadan MU sağlayan ve ilişkisel sınıflandırmaya dayanan yeni bir federe unutma yöntemi geliştirilmiştir. Sistem tasarımında, girdi ve çıktılar arasındaki ilişkilerin açıkça izlenebildiği ve şeffaflık sunan

ilişkisel sınıflandırma kuralları temel alınmıştır. Önerilen çerçevenin genel işleyişi ve sistem mimarisi Şekil 3.1. ile sunulmuştur.



Şekil 3.1. Önerilen çerçevenin genel işleyişi: (a) FedCBA ile kural üretimi ve birleştirme, (b) İstemci tabanlı unutma talebinin başlatılması, (c) Sunucu tarafında FedFORGET unutma stratejisinin uygulanması.

Şekil 3.1. ile FedCBA ve FedFORGET yöntemleri kullanılarak veriye erişmeden ve gizliliği koruyarak modelden veri silme işlemini gerçekleştiren bir yapı sunulmuştur. Sistem üç ana aşamadan oluşur:

Şekil 3.1. (a) FedCBA ile Kural Oluşturma ve Birleştirme: Önerilen sistemin ilk aşaması, Şekil 3.1. (a) bölümünde gösterildiği üzere federe öğrenmedeki birleştirme süreciyle başlar. Bu aşamada, sistemde yer alan yerel istemciler (İstemci 1, ..., N), kendi yerel veri setleri üzerinde CBA yöntemini kullanarak modellerini eğitirler. Bu modeller, verideki temel örüntüleri temsil eden etiketli ilişki kurallarından meydana gelmektedir. Eğitim tamamlandığında sunucu, bu yerel modelleri toplar ve kural tabanlı sınıflandırma için özel olarak tasarlanmış federe birleştirme algoritması olan FedCBA ile birleştirerek global modeli oluşturur ve güncellenmiş modeli tekrar istemcilere dağıtır. Bu süreç, sistem geliştiricisi tarafından belirlenen tur sayısına ulaşılana veya hedeflenen model performansı yakalanana kadar devam eder. Bu döngüsel yapı, gizlilik odaklı model eğitiminin temelini oluşturarak veriden habersiz bir öğrenme ortamı sağlar.

Şekil 3.1. (b) İstemci Düzeyinde Unutma İsteği: İkinci aşama, Şekil 3.1. (b) bölümünde gösterildiği gibi bir veya birden fazla istemcinin sistemden ayrılmak istemesi veya

verilerinin model üzerindeki etkisinin kaldırılmasını talep etmesiyle tetiklenir. FL mimarilerinde bu aşama; unutma taleplerinin hangi aralıklarla işleme alınacağı, taleplerin sıralama ve önceliklendirme mekanizması, taleplerin ve istemci model parametrelerinin depolanma formatı ve süresi gibi kritik mühendislik problemlerini beraberinde getirir. Ayrıca, istemcinin sistemdeki aktif veya pasif katkı düzeyinin unutma süreci üzerindeki etkisi ve sunucunun bu talebi doğrulama biçimi gibi zorluklar da bu kapsamda değerlendirilmektedir. Bu çalışma, söz konusu operasyonel ve tasarımsal sorunları doğrudan çözmekten ziyade, metodolojik doğruluğu merkeze alarak kural tabanlı federe unutma yöntemi geliştirmeye odaklanmıştır. Tüm bu mühendislik zorluklarının optimal şekilde yönetildiği varsayılarak gerçekleştirilen deneylerde, sunucu tarafındaki unutma işleminin istemcideki verilere ihtiyaç duymadan, veri egemenliğini koruyarak doğrudan model parametreleri üzerinden yürütülmesi sağlanmıştır.

Şekil 3.1. (c) FedFORGET Unutma Stratejisinin Uygulanması: Şekil 3.1. (c) bölümünde gösterilen son aşama, sunucu tarafındaki unutma işleminin fiilen yürütülmesini kapsar. Geçerli bir talep alındığında sunucu, ayrılan istemci tarafından sunulan modelleri mevcut birleştirilmiş global modelden matematiksel olarak çıkarır. Bu prosedür, tam bir yeniden eğitim döngüsüne ihtiyaç duyulmadan ve istemcinin verilerine maruz kalınmadan tamamlanır. İşlem sonucunda elde edilen unutmuş model, işlevsel olarak ayrılan istemcinin verilerine hiç maruz kalmamış bir modelle eşdeğer hale gelir. Sunucu, bu güncellenmiş modeli ağdaki diğer istemcilere geri dağıtarak sistem genelinde senkronizasyonu sağlar. Böylece, ilgili istemcinin etkisi tüm federe yapıdan silinmiş olur ve sistem, söz konusu veriyi hiç görmemiş gibi güncel parametrelerle işlemeye devam eder.

3.2. FedCBA Algoritmasının Tasarımı

Bu bölümde, çalışmanın temelini oluşturan FedCBA algoritmasının teknik mimarisi, kural üretim ve birleştirme süreçleri ile matematiksel altyapısı detaylandırılmaktadır.

3.2.1. Tanım ve mimari temeller

FedCBA, federe mimarisinde veri gizliliğini önceliklendirerek ilişkisel sınıflandırma yapabilen özgün bir federe birleştirme algoritmasıdır. Bu algoritma, temelde literatürde ilişkisel sınıflandırma yöntemi olan CBA algoritmasının FL mimarisine uyarlanmış ve optimize edilmiş bir versiyonudur. FedCBA, geliştirilme aşamasında daha önce önerilen Veriden habersiz ve artırımlı sınıflandırmaya dayalı birliktelik kuralları çıkarma mimarisi (Data Unaware Classification Based on Association – du-CBA) mimarisinden temel almıştır

[21]. Du-CBA, veriye erişim gerektirmeden istemcilerden gelen kural tabanlı modelleri sunucu tarafında birleştiren bir altyapı sunmaktadır.

CBA algoritması, veri setindeki öznitelikler ile sınıf etiketleri arasındaki ilişkileri "EĞER... İSE..." (IF... THEN...) yapısındaki kurallarla modellerken varsayılan olarak kural çıkarımı için Apriori algoritmasını kullanır. Ancak bu çalışmada, birliktelik kuralı madenciliği aşamasında farklı yaklaşımların performansa etkisi kapsamlı şekilde analiz edilmiştir. Apriori, Eclat ve FP-Growth algoritmaları üzerinde yapılan deneysel karşılaştırmalar sonucunda FP-Growth algoritmasının hem eğitim süresi verimliliği hem de sınıflandırma doğruluğu açısından en yüksek performansı sergilediği saptanmıştır. Bu bulgular doğrultusunda FedCBA, geleneksel du-CBA yapısını FP-Growth tabanlı bir kural üretim süreciyle entegre ederek daha optimize ve ölçeklenebilir bir model sunmaktadır.

3.2.2. Kural oluşturma ve birleştirme süreçleri

FedCBA algoritmasının işleyişi, FL prensiplerine uygun olarak iki temel fazda gerçekleşmektedir:

1. **İstemcilerde Yerel Eğitim Fazı (Local Training):** İlk olarak her bir istemci, kendi yerel veri setini dışarı sızdırmadan FP-Growth tabanlı CBA algoritması ile etiketli birliktelik kuralları üretir. Bu aşamada her kuralın geçerliliği, yerel destek ve güven metrikleriyle hesaplanmaktadır.
2. **Sunucu Tarafında Birleştirme (Aggregation):** İstemciler tarafından iletilen yerel modeller, sunucu üzerinde birleştirilerek global modelin oluşturulması sağlanır. Federe yapıda paylaşılan bu modeller özünde etiketli birliktelik kurallarından oluşmaktadır. İstemciler, yerel eğitim sonucunda elde ettikleri bu kurallar ile birlikte, kurallara ait destek ve güven değerleri ve yerel veri setlerinin örneklem boyutlarını sunucuya aktarır. Sunucu tarafındaki birleştirme aşamasında, gelen tüm kurallar karşılaştırmalı bir analize tabi tutularak işlenir. Aynı öncül ve aynı etikete sahip kurallar için destek ve güven değerleri geliştirilen formüller kullanılarak güncellenir. Aynı öncüle sahip ancak farklı etiketlere sahip kurallar arasında frekans kontrolü yapılır ve yüksek güven değerine sahip kural korunur. Eşleşme olmayan yeni kurallar global modele doğrudan dahil edilir.

3.2.3. Matematiksel formüller

Bu bölümde, FedCBA algoritması kapsamında kullanılan metrikler ve bu metriklerin federe ortamda nasıl birleştirildiği matematiksel olarak açıklanmaktadır. Öncelikle istemci

tarafında hesaplanan yerel metrikler tanımlanmakta, ardından sunucu tarafında gerçekleştirilen birleştirme işlemi formülize edilmektedir.

Her istemci, kendi yerel veri setini kullanarak etiketli birliktelik kuralları üretir. Üretilen bu kuralların geçerliliği ve güvenilirliği, destek ve güven metrikleri ile ölçülmektedir. Her istemci için hesaplanan destek formülü Denklem (4)'te, güven formülü Denklem (5)'te yer almaktadır.

$$Destek_i = \frac{Frekans_i(X,\ Y)}{N_i},\ \ i = 1, 2, 3, \dots, n \tag{4}$$

$$Güven_i = \frac{Frekans_i(X,\ Y)}{Frekans_i(X)},\ \ i = 1, 2, 3, \dots, n \tag{5}$$

Destek bir kuralın veri seti içerisinde ne sıklıkla gerçekleştiğini gösteren ve kuralın yaygınlığını ölçen istatistiksel bir metriktir. Güven ise bir kuralın doğruluğunu ifade eden ve öncül gerçekleştiğinde sonucun gerçekleşme olasılığını gösteren koşullu olasılık temelli bir metriktir. Denklem (4)'de i istemci indisini temsil etmektedir. i istemcisi ile tanımlanan ilgili istemci için *X* ve *Y* sırasıyla değerlendirilen kuralın öncül ve sonuç bileşenlerini *X →Y'*yi temsil etmektedir. N_i, i. istemcinin veri setindeki toplam örnek sayısını ifade etmektedir. $Frekans_i(X, Y)$, i. istemcinin *X* ve *Y*'nin birlikte görülme sıklığını ve $Frekans_i(X)$ ise i. istemcide *X*'in tek başına görülme sıklığını ifade etmektedir.

Sunucu tarafında gerçekleştirilen model birleştirme süreci, istemcilerden gelen kuralların ve bu kurallara ait metriklerin bütünleştirilmesini kapsamaktadır. Bu kapsamda, her istemci tarafından üretilen model, ilgili kurallar, bu kurallara ait yerel destek ve güven değerleri ile birlikte istemcinin örnek sayısı sunucuya iletilir. Sunucu, farklı istemcilerden gelen kuralları karşılaştırarak aynı kurala sahip olanların metriklerini günceller.

Denklem (4) ve Denklem (5)'e göre, aynı kurala sahip birden fazla istemci için (*n*) destek ve güven değerlerinin hesaplanmasında bazı büyüklükler doğrudan bilinirken, bazıları dolaylı olarak elde edilmektedir. Sunucuya iletilen destek, güven ve örnek sayısı değerleri bilinen büyüklükleri oluştururken, frekans bilgileri doğrudan paylaşılmamaktadır. Bu nedenle, aynı $X \rightarrow Y$ kuralına sahip tüm istemciler için bu bilinen değerler kullanılarak eksik bilgiler türetilmekte ve birleştirme işlemi gerçekleştirilmektedir.

Bu doğrultuda, farklı istemcilerden elde edilen aynı kurala ait metriklerin birleştirilmesiyle hesaplanan yeni destek ve güven değerleri sırasıyla Denklem (6) ve Denklem (7)'deki gibi tanımlanmaktadır.

$$Destek_{birleştirilmiş} = \frac{\sum_{i=1}^{n} Destek_i \; x \; N_i}{\sum_{i=1}^{n} N_i} \tag{6}$$

$$Güven_{birleştirilmiş} = \frac{\sum_{i=1}^{n} Destek_i \; x \; N_i}{\sum_{i=1}^{n} \frac{Destek_i \; x \; N_i}{Güven_i}} \tag{7}$$

Bu birleştirme işlemi, farklı istemcilerden gelen ve aynı etikete sahip kurallar için tekrarlanmaktadır. Bununla birlikte, farklı modellerde yer alan kuralların sol tarafı aynı olmasına rağmen sağ tarafı farklı ise, bu durum bir çakışma olarak değerlendirilir. Bu durumda, daha yüksek frekansa sahip olan kural korunarak modeldeki tutarlılık sağlanır.

3.2.4. FedCBA sözde kodu

Bu bölümde, FedCBA algoritmasının eğitim ve birleştirme sürecini açıklayan sözde kod (pseudocode) sunulmaktadır.

Algoritma 1. FedCBA Federe Birleştirme Algoritması: Eğitim ve Birleştirme Süreci

---

Girdi: İstemci veri setleri $D^{(1)}, D^{(2)}, \ldots, D^{(n)}$, destek eşiği *d*, güven eşiği *g*
Çıktı: Global model M_{global} – Kural Kümesi

1. Boş bir global model oluştur: $M_{global} \leftarrow \emptyset$
2. Toplam örnek sayısını başlat: $Toplam_Ornek \leftarrow 0$
3. Her istemci i=1, 2, ..., n için tekrarla:
   3.1. Yerel Eğitim Aşaması
   - $D^{(i)}$ veri setini işlem formatına dönüştür.
   - Yerel veri üzerinde CBA algoritmasını eğit:
     $M^{(i)}_{lokal} \leftarrow CBA(D^{(i)}, d, g)$
   - İstemci veri setindeki örnek sayısını belirle:
     $N_i \leftarrow /D^{(i)}/$
   - Toplam örnek sayısını güncelle:
     $Toplam_Ornek \leftarrow Toplam_Ornek + N_i$

   3.2. Sunucu Tarafı Birleştirme Aşaması

   - $M^{(i)}_{lokal}$ r_{yerel} için tekrarla:
     - $eşleşme \leftarrow yanlış$
     - M_{global} içindeki her global kural r_{global} için tekrarla:

       Aynı kural durumu:
       Eğer $r_{yerel}.lhs = r_{global}.lhsr$ *ve* $r_{yerel}.rhs = r_{global}.rhsr$ ise:

- r_{global} kuralının destek değeri $Destek_{birleştirilmiş}_Güncelle()$ fonksiyonu ile güncellenir.
- r_{global} kuralının güven değeri $Güven_{birleştirilmiş}_Güncelle()$ fonksiyonu ile güncellenir.
- *eşleşme ← doğru*
- İç döngü sonlandırılır.

Çakışan kural durumu:
Aksi halde, Eğer $r_{yerel}.lhs = r_{global}.lhsr$ *ve* $r_{yerel}.rhs \neq r_{global}.rhsr$ ise:

- Eğer $r_{yerel}.frekans > r_{global}.frekans$ ise:
  - r_{global}, r_{yerel} ile değiştirilir.
  - İlgili metrikler yeni frekansa göre güncellenir.
- *eşleşme ← doğru*
- İç döngü sonlandırılır.

- Eğer *eşleşme = yanlış* ise:
  - R_{yerel} kuralı M_{global} modeline eklenir.

4. M_{global} içindeki kurallar önce güven değerine, ardından destek değerine göre azalan biçimde sıralanır.
5. M_{global} döndürülür.

Algoritma 1'de yer alan FedCBA sözde koduna göre, FedCBA algoritması başlangıçta boş bir global modelin oluşturulması ve toplam örnek sayısının sıfır olarak başlatılması ile başlar. Bu aşama, tüm istemcilerden elde edilecek bilgilerin bir araya getirileceği temel yapının hazırlanmasını sağlar. Ardından, sistemde yer alan her istemci için işlemler tekrarlanır ve süreç iki ana aşamada yürütülür. Bunlar yerel eğitim ve birleştirme aşamalarıdır.

Yerel eğitim aşamasında, her istemci kendi veri setini kullanarak bağımsız bir model oluşturur. Bu kapsamda, istemciye ait veri seti birliktelik kuralı çıkarımı için uygun işlem formatına dönüştürülür ve CBA algoritması uygulanarak yerel kural kümesi elde edilir. Aynı zamanda, her istemcinin veri büyüklüğü hesaplanarak toplam örnek sayısına eklenir. Bu süreçte herhangi bir veri paylaşımı gerçekleştirilmez. Yalnızca model bilgisi üretilir ve bu durum veri gizliliğinin korunmasını sağlar.

Birleştirme aşaması sunucu tarafında gerçekleştirilir ve istemcilerden gelen modeller yani kuralların birleştirilmesini kapsar. Bu aşamada, her istemciden gelen yerel kurallar tek tek incelenir. Eğer bir yerel kural ile global modelde bulunan bir kuralın hem öncül hem de sonuç kısmı aynı ise ($r_{yerel}.lhs = r_{global}.lhsr$ *ve* $r_{yerel}.rhs = r_{global}.rhsr$), bu kurallar aynı kabul edilir ve ilgili destek ile güven değerleri güncellenir. Bu güncelleme işlemi $Destek_{birleştirilmiş}_Güncelle\ ()$ ve $Güven_{birleştirilmiş}_Güncelle\ ()$ fonksiyonları aracılığıyla gerçekleştirilir. Bu fonksiyonlar, farklı istemcilerden gelen aynı kural katkılarını bir araya getirerek global modele ait güncellenmiş destek ve güven değerlerini üretmektedir. Değerler

Denklem (6) ve Denklem (7)'de yer alan formüller ile elde edilmektedir. Bu noktada $Destek_{birleştirilmiş}_Güncelle\ ()$ ve $Güven_{birleştirilmiş}_Güncelle\ ()$ fonksiyonları, kural seviyesinde katkıların bütünleştirilmesini sağlayan ve global modelin oluşturulmasında temel rol oynayan fonksiyonel bileşenlerdir.

Eğer kuralların öncül kısmı aynı olmasına rağmen sonuç kısmı farklı ise ($r_{yerel}.lhs = r_{global}.lhsr$ ve $r_{yerel}.rhs \neq r_{global}.rhsr$), bu durum bir çakışma olarak değerlendirilir. Bu tür durumlarda, daha yüksek frekansa sahip olan kural tercih edilir ve global model bu kurala göre güncellenir. Böylece, sistemde baskın olan bilginin korunması sağlanır. Öte yandan, yerel modelde bulunan bir kural global modelde herhangi bir kuralla eşleşmiyorsa, bu kural yeni bilgi olarak kabul edilir ve doğrudan global modele eklenir.

Tüm istemcilerden gelen kurallar işlendiğinde, elde edilen global modeldeki kurallar öncelikle güven değerine, ardından destek değerine göre azalan biçimde sıralanır. Bu sıralama, sınıflandırma aşamasında daha güçlü kuralların öncelikli olarak kullanılmasını sağlar. Son olarak, oluşturulan global model çıktı olarak sunulur. Bu model, tüm istemcilerden elde edilen bilgilerin birleştirilmesiyle oluşturulmuş, açıklanabilir ve doğrulanabilir bir kural tabanlı yapıyı temsil etmektedir.

3.3. FedFORGET: Federe Mimaride Makine Unutması Yaklaşımı

Bu bölümde, FL sistemlerinde istemci düzeyinde veri silme taleplerini karşılamak amacıyla geliştirilen FedFORGET isimli federe unutma algoritmasının teknik tasarımı, kural tabanlı çıkarma mekanizması ve ilgili matematiksel formülasyonlar açıklanmaktadır.

3.3.1. Tanım ve mimari temeller

FedFORGET, FL ortamları için tasarlanmış, veri erişimi gerektirmeyen ve yaklaşık unutma prensibiyle çalışan bir algoritmadır. Sıfırdan tekrar eğitme yaklaşımı, bir istemci ayrıldığında yüksek hesaplama maliyeti oluştururken FedFORGET, global modeldeki ilişkisel kuralları algoritmik olarak güncelleyip bu maliyeti minimize eder. Mimari, kural tabanlı yapısı sayesinde kendinden yorumlanabilir/açıklanabilir özelliğine sahiptir ve unutma işleminin model üzerindeki etkisi doğrudan izlenebilir.

3.3.2. Kural çıkartma süreçleri

FedFORGET, FedCBA algoritması ile oluşturulan kural listesi üzerinden geriye dönük bir ayrıştırma işlemi yürütür. Süreç, sunucu tarafında şu teknik adımlarla gerçekleştirilir:

Talebin Karşılanması: Bir istemci sistemden ayrılma talebi gönderdiğinde, sunucu bu istemciye ait kural kümesini ve destek, güven, örneklem sayısı metriklerini girdi olarak alır. Bu aşamada izlenecek unutma stratejisi, sistem tasarımına bağlı olarak aktif veya pasif biçimde gerçekleştirilebilir.

Pasif unutma yaklaşımında, istemci sistemden çıkarıldıktan sonra modele doğrudan bir müdahale yapılmaz. İlgili istemcinin katkısı, yalnızca sonraki eğitim ve birleştirme süreçlerinde yer almaması nedeniyle zamanla etkisini yitirir. Ancak bu yaklaşımda, istemcinin geçmiş katkıları model içerisinde örtük olarak varlığını sürdürebilir ve tam anlamıyla bir unutma garantisi sağlanamaz.

Aktif unutma yaklaşımında ise, istemciye ait katkılar sunucuda önceden saklanan model bileşenleri kullanılarak doğrudan ve kontrollü bir şekilde modelden çıkarılır. Bu süreçte istemcinin yeniden sisteme dahil edilmesine gerek yoktur. Unutma işlemi tamamen sunucu tarafında gerçekleştirilen güncellemeler ile sağlanır.

Her iki yaklaşımın uygulanabilirliği sistem mimarisine bağlı olmakla birlikte, pasif unutma yalnızca istemci katkısının etkisini azaltırken, aktif unutma bu katkının modelden açık ve doğrulanabilir biçimde kaldırılmasını mümkün kılar. Bu bağlamda, FedFORGET, aktif unutma senaryoları için tasarlanmış olup, istemciye ait bilginin modelden izlenebilir ve ölçülebilir şekilde çıkarılmasını sağlar.

Kural Eşleşme Analizi: Global modeldeki her bir kural r_{global}, ayrılmak isteyen istemcinin yerel kurallarıyla r_{yerel} karşılaştırılır.

Parametrik Güncelleme: Unutulması talep edilen istemciye ait modelde yer alan bir kural, birden fazla istemci tarafından destekleniyorsa, söz konusu kuralın destek ve güven değerleri, ayrılan istemcinin katkısı dikkate alınarak orantılı biçimde matematiksel olarak güncellenir. Buna karşılık, yalnızca ilgili istemci tarafından üretilmiş ve diğer istemciler tarafından desteklenmeyen kurallar, global modelden tamamen ve kalıcı olarak çıkarılır.

3.3.3. Matematiksel formüller

Bu bölümde, FedFORGET yaklaşımı kapsamında kullanılan metrikler ve bu metriklerin unutma süreci doğrultusunda nasıl güncellendiği matematiksel olarak açıklanmaktadır. Öncelikle istemci tarafında hesaplanan yerel metrikler tanımlanmakta, ardından sunucu tarafında gerçekleştirilen kural tabanlı birleştirme ve istemci katkısının modelden çıkarılmasına yönelik parametrik güncelleme süreci formülize edilmektedir.

Unutma süreci, silme talebinde bulunan istemciye ait kuralların global model içerisinde var olup olmadığının kontrol edilmesi ile başlar. Global modelde bulunmayan kurallar için herhangi bir işlem yapılmazken, eşleşen kurallar için bir güncelleme süreci başlatılır. Bu aşamada, kuralın sahiplik durumuna bağlı olarak iki farklı senaryo ortaya çıkmaktadır. Eğer ilgili kural federe yapıdaki birden fazla istemci tarafından oluşturulmuşsa (ortak kural), kural modelden silinmez; ancak destek ve güven değerleri, ayrılan istemcinin katkısı çıkarılacak şekilde güncellenir. Bu güncelleme sonucunda kuralın frekansı ve global model içerisindeki öncelik sırası azalır. Buna karşılık, kural yalnızca sistemden ayrılan istemciye ait ise, güncelleme süreci sonucunda kuralın destek ve güven değerleri sıfıra veya önceden tanımlanmış bir eşik değerin altına düşer. Bu durumda ilgili kural geçersiz kabul edilerek global modelden tamamen ve kalıcı olarak çıkarılır. Söz konusu güncelleme işlemi, Denklem (8) ve Denklem (9) kullanılarak gerçekleştirilmektedir.

$$Destek_{unutturulmuş} = \frac{Frekans_t(X,\ Y) - Frekans_a(X,\ Y)}{N_t - N_a} \tag{8}$$

$$Güven_{unutturulmuş} = \frac{Frekans_t(X,\ Y) - Frekans_a(X,\ Y)}{Frekans_t(X) - Frekans_a(X)} \tag{9}$$

Denklem (8) ve Denklem (9)'da, *X* ve *Y* sırasıyla değerlendirilen kuralın öncül ve sonuç bileşenlerini, $X \rightarrow Y$'yi temsil ederken, *Frekans (X)* ve *Frekans (X, Y)* fonksiyonu ilgili kuralın veri setindeki frekansını ifade etmektedir. *N* değeri toplam örneklem büyüklüğünü gösterirken, alt indislerden *t* mevcut global modele ait değerleri, *a* ise sistemden ayrılmak isteyen istemciye ait değerleri temsil etmektedir.

Unutma işlemi hem global modelde hem de ayrılmak isteyen istemcinin yerel modelinde bulunan ortak $X \rightarrow Y$ kuralları üzerinden gerçekleştirilmekte olup, bu süreç istemcinin yerel modelinde yer alan her bir kural için ayrı ayrı uygulanmaktadır.

Buna ek olarak, $Frekans_t(X) - Frekans_a(X) = 0$ durumunda, ilgili kuralın yalnızca ayrılan istemciye ait olduğu kabul edilmekte ve bu tür kurallar global modelden doğrudan ve kalıcı olarak çıkarılmaktadır. Ayrıca, hesaplama sürecinde oluşabilecek negatif veya tanımsız değerlerin önüne geçmek amacıyla, metrik güncellemeleri uygun eşik ve kontrol mekanizmaları ile sınırlandırılmaktadır.

3.3.4. FedFORGET sözde kod

Bu bölümde, FedFORGET algoritmasının unutma sürecini açıklayan sözde kod (pseudocode) sunulmaktadır.

Algoritma 2. FedFORGET Federe Unutma Algoritması: Sunucu Tarafında Global Modelden İstemci Katkısının Çıkarılması Süreci

---

Girdi:
R_global : Global modele ait kural kümesi
R_ayrılan : Sistemden ayrılmak isteyen istemciye ait kural kümesi
N_toplam : Global modele ait toplam örneklem sayısı
N_ayrılan : Ayrılan istemciye ait örneklem sayısı

Çıktı:
R_yeni : Unutma işlemi sonrası güncellenmiş global kural kümesi
N_yeni : Unutma işlemi sonrası güncellenmiş örneklem sayısı

Her bir *r_ayrılan* ∈ *R_ayrılan* için:
Her bir *r_global* ∈ *R_global* için:
Eğer *r_global.öncül* = *r_ayrılan.öncül*
VE *r_global.sonuç* = *r_ayrılan.sonuç* ise:

destek_global ← *r_global.destek*
destek_ayrılan ← *r_ayrılan.destek*
güven_global ← *r_global.güven*
güven_ayrılan ← *r_ayrılan.güven*

yeni_destek ← $Destek_{unutturma}$*_Güncelle(*
destek_global, destek_ayrılan, N_toplam, N_ayrılan
)

yeni_güven ← $Güven_{unutturma}$*_Güncelle(*
destek_global, destek_ayrılan, güven_global, güven_ayrılan,
N_toplam, N_ayrılan
)

Eğer *yeni_destek* > *0* ise:
r_global.destek ← *yeni_destek*
r_global.güven ← *yeni_güven*
Aksi halde:
r_global.destek ← *0*
r_global.güven ← *0*
r_global kuralını *R_global* kümesinden çıkar
Bitti
Bitti
Bitti
Bitti

R_global kural kümesini önce güven değerine, ardından destek değerine göre azalan sırada sırala

N_yeni ← *N_toplam* − *N_ayrılan*

R_yeni ← *R_global*

R_yeni ve N_yeni değerlerini döndür

Algoritma 2, FedFORGET yaklaşımı kapsamında sunucu tarafında gerçekleştirilen federe unutma sürecini göstermektedir. Algoritmanın temel amacı, sistemden ayrılmak isteyen istemcinin global modele olan katkısını ham veriye ihtiyaç duymadan modelden çıkarmaktır. Bu işlem, istemciye ait yerel kural kümesi ile global modelde yer alan kuralların karşılaştırılması yoluyla gerçekleştirilir.

İlk aşamada, ayrılan istemcinin yerel modelinde bulunan her bir kural $r_{ayrılan}$, global modeldeki kurallar r_{global} ile karşılaştırılır. Burada $r_{ayrılan}$, sistemden ayrılan istemcinin katkısını temsil eden kuralı, r_{global} ise global modelde mevcut olan karşılık gelen kuralı ifade etmektedir. Karşılaştırma işlemi, kuralların öncül (*r.lhs*) ve sonuç (*r.rhs*) bileşenlerinin birebir eşleşmesine dayanır. Yani $r_{global}.öncül = r_{ayrılan}.öncül$ ve $r_{global}.sonuc = r_{ayrılan}.sonuc$ koşulları sağlandığında iki kuralın aynı bilgiye karşılık geldiği kabul edilir. Bu eşleşme sağlandığında, ilgili kural için global ve ayrılan istemciye ait parametreler ayrı ayrı atanır. Bu kapsamda global modeldeki destek ve güven değerleri $destek_{global}$, $güven_{global}$, ayrılan istemciye ait değerler ise $destek_{ayrılan}$ ve $güven_{ayrılan}$ olarak belirlenir. Ardından, bu değerler kullanılarak yeni destek ve güven değerleri hesaplanır. Bu hesaplama işlemi doğrudan algoritma içinde yapılmak yerine, $Destek_{unutturma}_Güncelle\ ()$ ve $Güven_{unutturma}_Güncelle\ ()$ fonksiyonları aracılığıyla gerçekleştirilir. Bu fonksiyonlar, ilgili kuralın global modeldeki toplam katkısından ayrılan istemcinin katkısını çıkararak güncellenmiş değerleri üretmektedir. Bu işlemler Denklem (8) ve Denklem (9)'da yer alan ve çalışma kapsamında geliştirilen formüller ile gerçekleştirilir. Denklem (8)'de yer alan formül, global destek değerinin toplam katkısını temsil eden paydan, ayrılan istemcinin katkısını çıkararak kalan katkının güncellenmiş örneklem sayısına normalize edilmesini sağlamaktadır. Benzer şekilde Denklem (9)'da yer alan formül, güven değeri de doğrudan çıkarım ile değil, güncellenmiş destek değerleri üzerinden yeniden hesaplanmaktadır. Bu noktada $Destek_{unutturma}_Güncelle\ ()$ ve $Güven_{unutturma}_Güncelle\ ()$ fonksiyonları, ilgili matematiksel işlemlerin gerçekleştirildiği temel bileşenler olup, katkı çıkarımının doğrudan uygulandığı yerdir.

Elde edilen *yeni_destek* değeri üzerinden bir kontrol mekanizması çalıştırılmaktadır. Eğer $yeni_destek > 0$ ise, bu durum ilgili kuralın yalnızca ayrılan istemciye ait olmadığını, diğer istemcilerin de bu kurala katkı sağladığını göstermektedir. Bu durumda kural global modelde tutulur ve yalnızca destek ve güven değerleri güncellenerek etkisi azaltılır. Buna karşılık, $yeni_destek = 0$ olması durumunda, kuralın yalnızca ayrılan istemci tarafından üretildiği ve başka hiçbir istemci tarafından desteklenmediği anlaşılır. Bu nedenle ilgili kural geçersiz kabul edilerek global modelden tamamen çıkarılır.

Son aşamada, güncellenmiş global kural kümesi, modelin karar verme sürecinde daha güçlü kuralların öncelikli olarak değerlendirilmesini sağlamak amacıyla, önce güven değerine ardından destek değerine göre azalan sırada yeniden sıralanır. Ayrıca toplam örneklem sayısı $N_{yeni} = N_{toplam} - N_{ayrılan}$ olacak şekilde güncellenir. Böylece sistem, yeniden eğitim gerektirmeden, ayrılan istemcinin modele olan etkisinden arındırılmış güncel global modeli elde eder.

4. DENEYSEL ÇALIŞMALAR VE BULGULAR

Bu bölümde, geliştirilen FedCBA ve FedFORGET algoritmalarının hem istatistiksel başarımını hem de sistem üzerindeki uygulama verimliliğini değerlendirmek amacıyla gerçekleştirilen deneysel çalışmalar ve elde edilen sonuçlar sunulmaktadır. Çalışma kapsamında öncelikle deneylerde kullanılan veri setlerinin nitelikleri ve modellerin etkinliğini ölçmek için belirlenen değerlendirme kriterleri açıklanmaktadır. İlerleyen kısımlarda, FedCBA algoritmasının sınıflandırma performansı ve FedFORGET yönteminin farklı senaryolar dahilindeki unutma başarısı zaman ve doğruluk parametreleri ile birlikte

detaylandırılmaktadır. Ayrıca, geliştirilen yaklaşımların gerçek dünya koşullarındaki geçerliliğini test etmek amacıyla Google Cloud tabanlı dağıtık mimari üzerinde API haberleşme süreçleri ile performans analizleri ve Raspberry Pi gibi uç cihazlardaki uygulama sonuçları incelenmektedir.

4.1. Veri Setleri

Bu çalışmada, geliştirilen FedCBA federe birleştirme ve FedFORGET MU algoritmalarının başarımını, genellenebilirliğini ve farklı donanım mimarilerindeki uygulama verimliliğini değerlendirmek amacıyla literatürde yaygın olarak kabul gören veri setleri kullanılmıştır.

CBA algoritmasının birliktelik kuralı üretmesi sebebi ile sürekli (nümerik) verilerden ziyade ayrık (kategorik) veriler üzerinde anlamsal ilişkileri yakalayabildiği ve mantıksal kural çıkarımında daha başarılı olduğu bilinmektedir [197]. Nümerik verilerin kural tabanlı sistemlerde işlenmesi genellikle veri kaybına, kural karmaşasına ve uzun kural üretim sürelerine yol açabilmektedir. Bu sebeple, geliştirilen algoritmaların potansiyelini üst düzeye çıkarmak ve doğru bir değerlendirme yapabilmek adına, deneysel çalışmalarda kullanılmak üzere tamamı kategorik özniteliklere sahip veya kategorik hale getirilmiş veri setleri seçilmiştir.

Çalışmanın deneysel aşamalarında kullanılan veri setleri ve kullanım amaçları şu şekildedir:

Temel Referans Veri Seti: Tez kapsamında tüm deneysel çalışmaların temel referans noktasını oluşturmak üzere IBM İnsan Kaynakları Analitiği: Çalışan Yıpranması/İşten Ayrılma ve Performans veri seti (IBM HR Analytics Employee Attrition & Performance dataset, IBM) tercih edilmiştir [198]. Bu veri seti insan kaynakları alanında çalışan kaybını tahminlemeyi amaçlamaktadır. Kategorik özellikleri sebebi ile CBA'nın anlamlı ilişkiler çıkarmasını sağlamış ve CBA ile eğitilmiş modelleri birleştiren FedCBA algoritmasının da performansını ölçmek için uygun bir zemin sunmuştur.

Çalışma kapsamında FedCBA algoritmasının performans analizlerinde, Google Cloud tabanlı dağıtık bulut mimarisi ve Raspberry Pi mimarisi testlerinde yalnızca bu veri seti kullanılmıştır.

FedFORGET Analizleri İçin Veri Setleri: Verinin global modelden unutulmasını hedefleyen FedFORGET yaklaşımının genellenebilirliğini kanıtlamak amacıyla deney kapsamı genişletilmiştir. Bu doğrultuda, algoritmanın performansı yalnızca tek bir referans veri setiyle sınırlandırılmamıştır. Yöntemin tutarlı ve kararlı yapısını ortaya koymak için

ikili ve çok sınıflı yapıya sahip farklı veri setleri kullanılmıştır. Ayrıca, önerilen yaklaşımın IID ve Non-IID senaryoları altında da testleri gerçekleştirilmiştir.

FedFORGET algoritmasının unutma performans analizleri; IBM İnsan Kaynakları Analitiği veri seti, Adult (Census Income) veri seti [199], Mushroom veri seti [200], Cars Evaluation veri seti [201] ve Nursery veri seti [202] olmak üzere toplam beş veri seti üzerinde toplu olarak gerçekleştirilmiştir.

1. IBM İnsan Kaynakları Analitiği: Çalışan Yıpranması/İşten Ayrılma ve Performans Veri Seti:

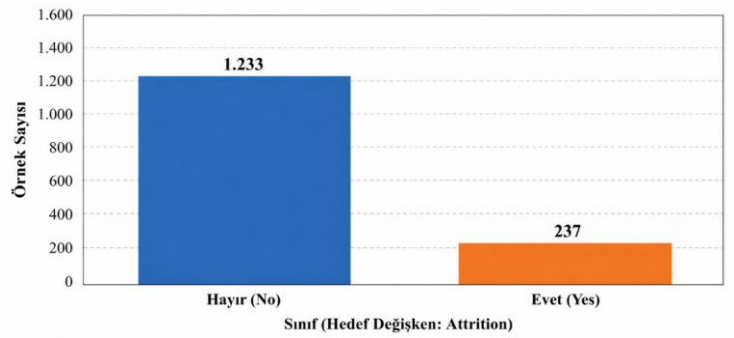
IBM İnsan Kaynakları Analitiği: Çalışan Yıpranması/İşten Ayrılma ve Performans Veri Seti, IBM tarafından geliştirilmiş ve sentetik olarak üretilmiş, açık erişimli bir veri setidir. Bu veri seti, çalışanların işten ayrılma davranışlarını ve performanslarını analiz etmeye yönelik çeşitli demografik ve kurumsal bilgileri içermektedir.

Veri seti; yaş, departman, maaş düzeyi, iş tatmini, çalışma süresi gibi birçok özelliği kapsamakta olup, özellikle insan kaynakları departmanlarının çalışan bağlılığını, işten ayrılma eğilimlerini ve performans dinamiklerini anlamalarına yardımcı olacak şekilde tasarlanmıştır. Bu yönüyle, çalışan yıpranması tahmini problemleri için literatürde yaygın olarak kullanılan bir veri setidir.

Tablo 4.1. IBM İnsan Kaynakları Analitiği veri setine ait temel özellikler sunulmaktadır. Tablo 4.1. incelendiğinde, veri setinin 1470 örnek ve 35 öznitelikten oluştuğu, hedef değişkenin "Attrition (Yes/No)" olduğu ve veri dağılımının dengesiz bir yapıya sahip olduğu görülmektedir. Şekil 4.1.'de IBM İnsan Kaynakları Analitiği veri setindeki hedef değişkenin sınıflara göre dağılımı yer almaktadır.

Tablo 4.1. IBM İnsan Kaynakları Analitiği veri setine ait temel özellikler.

Özellik	Değerler
Veri Seti Adı	IBM İnsan Kaynakları Analitiği
Veri Seti Kaynağı	[198]
Örneklem Sayısı	1470
Öznitelik Sayısı	35
Sınıf Sayısı (İkili / Çok Sınıflı)	2 (İkili)
Sınıf Dağılımı (Dengeli / Dengesiz)	Dengesiz
Hedef Öznitelik	Attrition (Yes / No)



Şekil 4.1. IBM İnsan Kaynakları Analitiği veri setindeki hedef değişkenin sınıflara göre dağılımı.

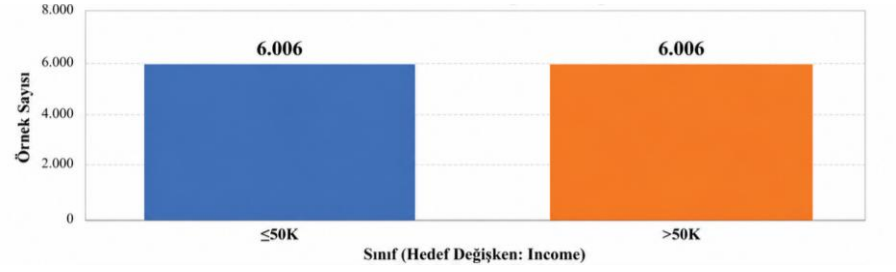
2. Adult (Census Income) Veri Seti:

Adult veri seti, bireylerin demografik ve yaş, eğitim düzeyi, meslek, medeni durum, haftalık çalışma süresi gibi sosyo-ekonomik özelliklerine dayanarak yıllık gelirlerinin 50.000 doların altında mı yoksa üstünde mi olduğunu tahmin etmeye yönelik klasik bir sınıflandırma problemidir. Veri seti açık erişimli olup literatürde gelir tahmini, sınıflandırma ve ayrımcılık analizi çalışmalarında kullanılmaktadır.

Bu çalışmada veri seti, FL mimarisine uygun olacak şekilde yeniden düzenlenmiş ve sınıf dengesizliğini ortadan kaldırmak amacıyla dengeli hale getirilmiştir. Tablo 4.2. ile Adult veri setine ait temel özellikler sunulmaktadır. Veri seti iki sınıftan oluşmakta olup hedef değişken "salary" olarak belirtilmiştir. Veri seti, her iki sınıf için eşit sayıda örnek içerecek şekilde dengelenmiş olup, her bir sınıf için 6.006 örnek bulunmaktadır. Şekil 4.2.'de dengelenmiş Adult veri setindeki hedef değişkenin sınıflara göre dağılımı yer almaktadır.

Tablo 4.2. Adult veri setine ait temel özellikler.

Özellik	Değer
Veri Seti Adı	Adult (Census Income)
Veri Seti Kaynağı	[199]
Örneklem Sayısı	12.012
Öznitelik Sayısı	14
Sınıf Sayısı (İkili / Çok Sınıflı)	2 (İkili)
Sınıf Dağılımı (Dengeli / Dengesiz)	Dengeli (6006 / 6006)
Hedef Öznitelik	Income (>50K / ≤50K)



Şekil 4.2. Adult veri setindeki hedef değişkenin dengelenmiş sınıf dağılımı.

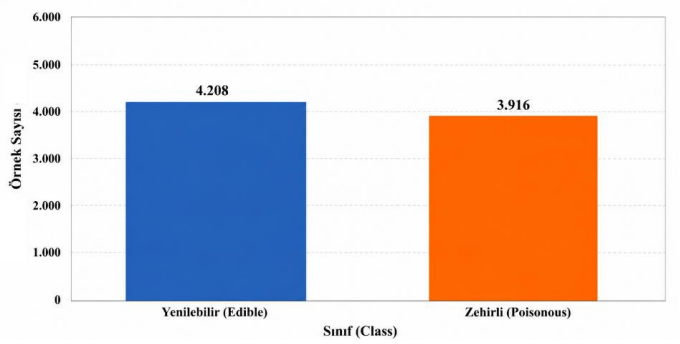
3. Mushroom Veri Seti:

Mushroom veri seti, mantarların morfolojik özelliklerine dayanarak yenilebilir veya zehirli olup olmadığını tahmin etmeye yönelik bir sınıflandırma veri setidir. Veri seti tamamen kategorik özniteliklerden oluşmakta olup, mantarın şapka yapısı, sap özellikleri, solungaç yapısı, renk ve koku gibi gözlemsel özelliklerini içermektedir. Bu veri seti, sınıflandırma algoritmalarının performansını değerlendirmek amacıyla literatürde temel veri setlerinden biridir.

Tablo 4.3. ile Mushroom veri setine ait temel özellikler sunulmaktadır. Veri seti toplamda 8.124 örnek ve 23 öznitelikten oluşmaktadır. Hedef değişken, mantarın yenilebilir veya zehirli olma durumunu ifade eden iki sınıflı bir yapıya sahiptir. Şekil 4.3.'te Mushroom veri setindeki hedef değişkenin sınıflara göre dağılımı yer almaktadır.

Tablo 4.3. Mushroom veri setine ait temel özellikler.

Özellik	Değer
Veri Seti Adı	Mushroom
Veri Seti Kaynağı	[200]
Örneklem Sayısı	8.124
Öznitelik Sayısı	23
Sınıf Sayısı (İkili / Çok Sınıflı)	2 (İkili)
Sınıf Dağılımı (Dengeli / Dengesiz)	Dengesiz (yaklaşık dengeli)
Hedef Öznitelik	Class (edible / poisonous)



Şekil 4.3. Mushroom veri setindeki hedef değişkenin sınıflara göre dağılımı.

4. Cars Evaluation Veri Seti:

Cars Evaluation veri seti, otomobil satın alma kararlarını değerlendirmek amacıyla geliştirilmiş klasik birçok sınıflı sınıflandırma veri setidir. Veri seti, araçların satın alma fiyatı, bakım maliyeti, kapı sayısı, kişi kapasitesi, bagaj büyüklüğü ve güvenlik düzeyi gibi

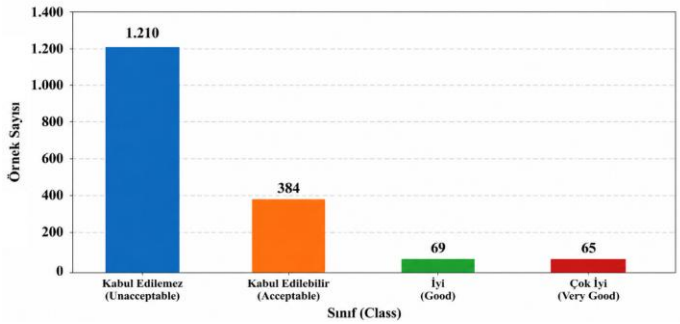
özelliklere dayanmaktadır. Bu veri seti karar verme sistemleri ve sınıflandırma algoritmalarının performansını değerlendirmek amacıyla literatürde kullanılmaktadır.

Veri seti, araçların genel değerlendirme durumunu dört farklı sınıfa ayırmaktadır: *unacc (kabul edilemez), acc (kabul edilebilir), good (iyi), vgood (çok iyi)*. Bu yönüyle çok sınıflı sınıflandırma problemleri için uygun bir yapı sunmaktadır.

Tablo 4.4. ile Cars Evaluation veri setine ait temel özellikler sunulmaktadır. Veri seti toplam 1.728 örnekten ve 6 öznitelikten oluşmaktadır. Hedef değişken dört sınıftan oluşmakta olup veri seti sınıf dağılımı açısından dengesiz bir yapıya sahiptir. Şekil 4.4.'te Cars Evaluation veri setindeki hedef değişkenin sınıflara göre dağılımı yer almaktadır.

Tablo 4.4. Cars Evaluation veri setine ait temel özellikler.

Özellik	Değer
Veri Seti Adı	Cars Evaluation
Veri Seti Kaynağı	[201]
Örneklem Sayısı	1.728
Öznitelik Sayısı	6
Sınıf Sayısı (İkili / Çok Sınıflı)	4 (Çok Sınıflı)
Sınıf Dağılımı (Dengeli / Dengesiz)	Dengesiz
Hedef Öznitelik	Class (unacc, acc, good, vgood)



Şekil 4.4. Cars Evaluation veri setindeki hedef değişkenin sınıflara göre dağılımı.

5. Nursery Veri Seti:

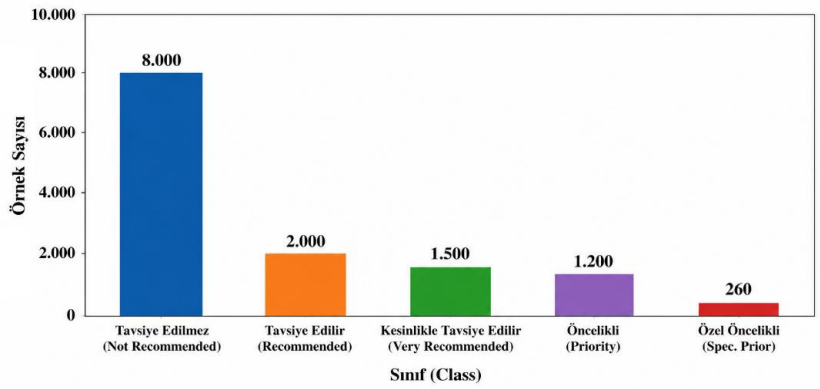
Nursery veri seti, çocukların kreş kabul sürecinde değerlendirilmesine yönelik geliştirilmiş çok sınıflı bir sınıflandırma veri setidir. Veri seti; aile yapısı, ebeveynlerin çalışma durumu, mali durum, sosyal koşullar ve çocuk sayısı gibi çeşitli sosyo-ekonomik özellikleri içermektedir. Bu veri seti özellikle karar destek sistemleri ve sınıflandırma algoritmalarının performans analizinde yaygın olarak kullanılmaktadır.

Veri seti, bireyleri beş farklı sınıfa ayırmaktadır: *not_recom (önerilmez), recommend (önerilir), very_recom (yüksek derecede önerilir), priority (öncelikli), spec_prior (özel öncelikli)*. Bu yönüyle çok sınıflı sınıflandırma problemleri için uygun bir yapı sunmaktadır.

Tablo 4.5. ile Nursery veri setine ait temel özellikler sunulmaktadır. Veri seti toplam 12.960 örnekten ve 8 öznitelikten oluşmaktadır. Hedef değişken beş sınıftan oluşmakta olup veri seti sınıf dağılımı açısından dengesiz bir yapıya sahiptir. Şekil 4.5.'te Nursery veri setindeki hedef değişkenin sınıflara göre dağılımı yer almaktadır.

Tablo 4.5. Nursery veri setine ait temel özellikler.

Özellik	Değer
Veri Seti Adı	Nursery
Veri Seti Kaynağı	[202]
Örneklem Sayısı	12.960
Öznitelik Sayısı	8
Sınıf Sayısı (İkili / Çok Sınıflı)	5 (Çok Sınıflı)
Sınıf Dağılımı (Dengeli / Dengesiz)	Dengesiz
Hedef Öznitelik	Label (not_recom, recommend, very_recom, priority, spec_prior)



Şekil 4.5. Nursery veri setindeki hedef değişkenin sınıflara göre dağılımı.

4.2. Değerlendirme Kriterleri

Bu çalışmada, geliştirilen FedCBA federe birleştirme ve FedFORGET unutma algoritmalarının performanslarını objektif ve çok boyutlu bir şekilde değerlendirebilmek için hem model performans metrikleri hem de verimlilik kriterleri kullanılmıştır. Deneysel sonuçların analizinde sırasıyla; Doğruluk (Accuracy), Kesinlik (Precision), Duyarlılık (Recall) ve F1-Skoru (F1-Score) model performans metrikleri kullanılmıştır [203].

FU süreçlerinde modelin başarısını değerlendirmek, geleneksel makine öğrenmesi yaklaşımlarından daha karmaşıktır. Modelin hem genel tahmin yeteneğini koruduğunu hem

de istenilen veriyi başarıyla sildiğini kanıtlamak için değerlendirmeler iki farklı veri seti üzerinden yürütülmektedir [91].

- Test Verisi Üzerindeki Performans: Test verisi, modelin eğitim sürecinde hiç görmediği ve genel sınıflandırma yeteneğini ölçmek için ayrılmış veri setidir. Başarılı bir unutma işleminden sonra, global modelin test verisi üzerindeki performans metriklerinin düşmeyip sabit kalması beklenmektedir. Bu durum, hedef veri silinirken modelin diğer verilerden öğrendiği genel kuralların ve faydalı bilgilerin bozulmadığını kanıtlar ve modelin genel başarımını gösterir.

- Hedef Veri (Target Data) Üzerindeki Performans: Hedef veri, sistemden unutulması talep edilen veridir. Model, başlangıçta bu veriyi görerek eğitildiği için unutma işleminden önce hedef veri üzerinde kısmen yüksek bir doğruluk oranına sahiptir. Unutma işlemi başarıyla gerçekleştiğinde, global modelin bu hedef veri üzerindeki performansının düşmesi gerekmektedir. Hedef veri üzerindeki bu performans düşüşü, modelin o veriye ait özel kuralları unuttuğunu ve söz konusu veriyi sanki hiç görmemiş gibi davrandığını ispatlayarak modelin unutma başarımını gösterir. İdeal bir unutma senaryosunda, unutmuş modelin hedef veri üzerindeki performansı, o veri hiç kullanılmadan sıfırdan eğitilmiş bir modelin performansına denk olmalıdır.

FU süreçlerinde temel motivasyon, global modelin yeniden eğitilmesi durumunda ortaya çıkan hesaplama maliyetlerini optimize etmektir. Bu bağlamda, çalışmada sistem verimliliğini değerlendirmek üzere eğitim süresi ve unutma süresi olmak üzere iki temel zaman metriği karşılaştırmalı olarak analiz edilmiştir. Eğitim süresi, FedCBA birleştirme algoritması aracılığıyla global modelin baştan oluşturulması aşamasında harcanan toplam zamanı ifade etmektedir. Unutma süresi ise, FedFORGET algoritması işletildiğinde, hedeflenen verinin model üzerindeki etkisinin ortadan kaldırılması süresince harcanan toplam süreyi tanımlamaktadır. Gerçekleştirilen deneylerde, FedFORGET algoritmasının sunduğu unutma süresi ile modelin sıfırdan eğitilmesi için gereken süre karşılaştırılarak, geliştirilen yaklaşımın sağladığı verimlilik nicel olarak ölçümlenmiştir.

4.3. FedCBA Performans Analizi ve Sonuçları

Bu bölümde, ilişkisel sınıflandırma tabanlı FedCBA federe birleştirme algoritmasının FL mimarisindeki performansı değerlendirilmektedir [1]. Değerlendirme kapsamında; deney

ortamının donanımsal ve yazılımsal kurulumu detaylandırılmakta, kullanılan veri seti ve veri seti üzerinden kurgulanan federe senaryo açıklanmaktadır. Elde edilen bulgular literatürdeki mevcut yaklaşımlarla karşılaştırmalı olarak sunulmaktadır.

4.3.1. Deney ortamı

Bu çalışmada gerçekleştirilen deneysel analizler, 11. Nesil Intel i5-1135G7 işlemciye ve 16 GB DDR3 belleğe sahip bilgisayar üzerinde yürütülmüştür. Geliştirme süreci için PyCharm entegre geliştirme ortamı tercih edilmiştir. Algoritmik kodlamalar Python programlama dili ve Python pyArc modülü kullanılarak gerçekleştirilmiştir. Geliştirilen algoritma gerçek dünya uygulamalarında kullanılmadan önce kontrollü bir şekilde test edilip onaylanması amacıyla, federe mimarisi simülasyon ortamı kurgulanmış ve tüm süreçler yerel bilgisayar ortamında simüle edilmiştir.

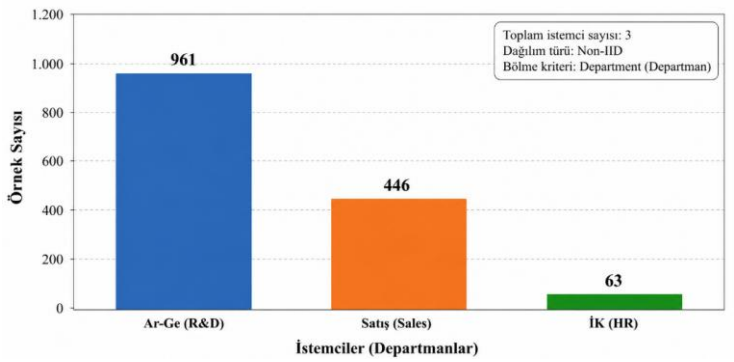
Simülasyon ortamının çalışma prensibi, eğitim verisinin istemci sayısı kadar bağımsız alt parçaya bölünmesi ve her bir parçanın ayrı bir yerel istemciyi temsil ettiği düşünülerek işlenmesine dayanmaktadır. Bu yapıda, veri setinin eğitim bölümü hedeflenen istemci sayısına göre ayrılmakta ve her bir istemci yalnızca kendi yerel verisini kullanarak CBA algoritması ile bağımsız bir yerel model eğitmektedir. Ardından elde edilen yerel modeller, FedCBA algoritması kullanılarak merkezi sunucuda birleştirilmektedir. Böylece, gerçek federe öğrenme ortamlarında ortaya çıkan ağ iletişim gecikmeleri ve veri iletim maliyetlerinden bağımsız olarak önerilen yaklaşımın çalışma performansı değerlendirilmiştir.

FedCBA algoritmasının performansını ölçmek amacıyla IBM İnsan Kaynakları Analitiği veri seti kullanılmış ve federe öğrenme ortamı bu veri seti üzerinde yapılandırılmıştır. Simülasyon sürecinde veri seti %80 eğitim ve %20 test verisi olarak ayrılmıştır. Eğitim verisi, gerçek dünya koşullarındaki heterojen (Non-IID) dağılımı yansıtmak amacıyla Departman özniteliğine göre üçe ayrılmıştır. Departman sütununda yer alan Satış, Araştırma ve Geliştirme (Ar-Ge) ve İnsan Kaynakları (İK) değerlerinin her biri ayrı bir istemciyi temsil etmektedir. Kurulan federe mimari simülasyonu kapsamında veri dağılımına ilişkin temel yapılandırma detayları Tablo 4.6.'da sunulmuş olup, istemcilere atanan veri dağılımı Şekil 4.6.'da görselleştirilmiştir.

Tablo 4.6. IBM İnsan Kaynakları Analitiği veri setinin federe mimari kapsamındaki veri dağılım parametreleri.

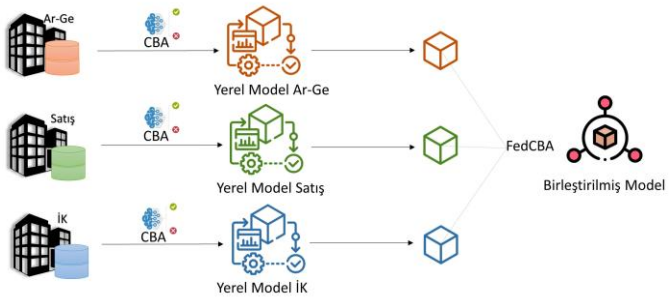
Özellik	Değerler
Bölme Kriteri	Department

İstemci Sayısı	3
İstemci İsimleri	Research & Development, Sales, Human Resources
Veri Dağılım Türü (IID / Non-IID)	Non-IID



Şekil 4.6. IBM İnsan Kaynakları Analitiği veri setinin federe öğrenme mimarisindeki istemcilere departman bilgisine göre Non-IID dağılımı

İstemcilerin yerel modellerini eğitip sunucuya ilettiği ve sunucuda FedCBA algoritması ile birleştirildiği bu iş akışı Şekil 4.7.'de sunulmaktadır. Algoritmaların kural çıkarım süreçlerinde, temel odak noktası doğru sınıflandırma yapmak olduğu için yalnızca destek ve güven metrikleri dikkate alınmıştır. Deneysel kurgunun optimizasyon aşamasında, pyArc modülü içerisindeki destek eşik değeri 0.01, güven eşik değeri ise 0.2 olarak belirlenmiş ve tüm yerel model eğitimleri bu parametreler üzerinden standartlaştırılarak gerçekleştirilmiştir.



Şekil 4.7. IBM İnsan Kaynakları Analitiği veri seti kullanılarak, farklı veri dağılımlarına sahip istemcilerde eğitilen yerel CBA modellerinin, FedCBA yaklaşımı ile merkezi sunucuda birleştirilerek global modelin oluşturulması süreci.

IBM İnsan Kaynakları Analitiği veri setinin model performansını artıracak şekilde hazırlanabilmesi amacıyla veri ön işleme adımları uygulanmıştır. Bu doğrultuda, veri setinde model başarımını olumsuz etkileyebilecek tekrar eden, eksik veya dengesiz veri gibi problemler ele alınmıştır.

Veri setinde sınıflandırma işlemi, “Attrition” özelliğine göre yapılmış olup bu özellik ikili kategorik bir değişken olarak “No” ve “Yes” değerlerini içermektedir. “No” sınıfı halen çalışmakta olan personeli, “Yes” sınıfı ise işten ayrılan personeli temsil etmektedir. Toplam 1.470 personelden 1.233’ünün çalışmaya devam ettiği, 237’sinin ise işten ayrıldığı gözlemlenmiştir.

Bu dağılım, veri setinde belirgin bir etiket dengesizliği bulunduğunu göstermektedir. Etiket dengesizliği, çoğunluk sınıfın baskın olduğu ve azınlık sınıfın yetersiz temsil edildiği durumları ifade etmekte olup, modelin çoğunluk sınıfa yönelmesine ve azınlık sınıfı yeterince öğrenememesine neden olabilmektedir. Özellikle FL gibi dağıtık ortamlarda her istemcinin farklı veri dağılımına sahip olması bu etkiyi daha da belirgin hale getirmekte, modelin genellenebilirliğini azaltmakta ve sınıflandırma performansını olumsuz yönde etkileyebilmektedir. Bu problemi gidermek amacıyla, kuralların yorumlanabilirliğini korumak için Random Oversampling [204] yöntemi tercih edilmiştir. Bu yöntem ile azınlık sınıfa ait örnekler çoğaltılarak veri setindeki sınıf dağılımı dengelenmiştir. Uygulama sonrasında “No” ve “Yes” sınıflarına ait örnek sayıları eşitlenmiş ve toplam örnek sayısı 2.466’ya ulaşmıştır.

Dengeleme işleminin ardından, kategorik değişkenler arasındaki anlamlı ilişkileri belirlemek amacıyla Ki-kare testi uygulanmıştır [205]. Bu test, kategorik değişkenler arasındaki bağımsızlığı değerlendirme gücü nedeniyle tercih edilmiş olup, istatistiksel olarak anlamlı ilişki göstermeyen “EmployeeCount”, “Over18”, “PerformanceRating” ve “StandardHours” değişkenleri veri setinden çıkarılmıştır. Ayrıca, Python pandas kütüphanesi kullanılarak veri setinde eksik ve tekrar eden kayıtlar kontrol edilmiş ve herhangi bir eksik ya da yinelenen veri bulunmadığı tespit edilmiştir.

4.3.2. Deneysel bulgular ve performans karşılaştırması

du-CBA, sunucu tarafında çalışan ve federe öğrenme mimarisi içerisinde model birleştirme işlemini gerçekleştiren bir federe birleştirme algoritmasıdır [21]. Bu algoritma modele özgü yaklaşım benimsemekte olup, istemci tarafında CBA algoritması ile eğitilmiş yerel modellerin bir araya getirilerek global modelin oluşturulmasını sağlamaktadır. CBA algoritması, sınıflandırma amacıyla etiketli birliktelik kuralları üretmektedir. Birliktelik kurallarının çıkarılmasında varsayılan olarak Apriori algoritması kullanılmaktadır. Bu çalışmada, Apriori algoritmasına ek olarak Eclat ve FP-Growth algoritmaları da CBA yapısına entegre edilmiş ve performansları analiz edilmiştir.

Tablo 4.7.'de, birliktelik kuralı algoritmalarının CBA modeli içerisindeki performans karşılaştırması sunulmaktadır. Bu kapsamda, Apriori algoritması varsayılan yöntem olarak kullanılmış, ardından Eclat ve FP-Growth algoritmaları CBA yapısına sırasıyla entegre edilerek aynı veri seti üzerinde test edilmiştir. Her bir algoritma, aynı deneysel koşullar altında çalıştırılmış ve elde edilen doğruluk, kesinlik, duyarlılık, F1-skor ve eğitim süresi metrikleri karşılaştırmalı olarak değerlendirilmiştir.

Tablo 4.7. Birliktelik kuralı algoritmalarının performans karşılaştırması.

Algoritma	Doğruluk (%)	Kesinlik (%)	Duyarlılık (%)	F1-Skor (%)	Eğitim Süresi (Saniye)
CBA_Apriori	84	84	84	84	29.46
CBA_Eclat	83	84	84	83	31.01
CBA_FP-Growth	87	87	87	87	25.30

Dengeli veri seti ve yaklaşık IID dağılım üzerinde gerçekleştirilen deneyler sonucunda aşırı öğrenme problemi gözlemlenmemiş ve analizler birliktelik kural algoritmalarının karşılaştırılması yönünde devam etmiştir. Varsayılan yöntem olan Apriori algoritması ile başlayan süreçte, aynı adımlar Eclat ve FP-Growth algoritmaları için de uygulanmış ve her bir algoritmanın model performansına etkisi değerlendirilmiştir. Elde edilen sonuçlara göre Apriori algoritması tüm metriklerde ortalama %84 başarı göstermiştir. Eclat algoritması benzer sonuçlar üretmekle birlikte özellikle eğitim süresi açısından daha uzun ve değişken bir performans sergilemiştir. FP-Growth algoritması ise metriklerinin tamamında %87 ortalama değer ile en iyi performansı göstermiştir. Ayrıca ortalama 25.30 saniyelik eğitim süresi ile diğer algoritmalara kıyasla daha hızlı çalışarak zaman açısından da avantaj sağlamıştır.

CBA algoritmasında varsayılan olarak kullanılan Apriori yönteminin FP-Growth algoritması ile değiştirilmesi, daha hızlı ve daha doğru birliktelik kurallarının elde edilmesini mümkün kılmıştır. FP-Growth algoritmasının CBA yapısına entegre edilmesi, du-CBA algoritmasının performansında önemli iyileştirmeler sağlamış ve sınıflandırma doğruluğunu artırmıştır. İşleme hızı ve doğruluk açısından sağladığı üstünlük sayesinde FP-Growth tabanlı yaklaşım, federe öğrenme ortamında daha etkin sonuçlar üretmiştir. Bu kapsamda, CBA algoritmasında yapılan iyileştirmelerin du-CBA algoritmasına entegre edilmesiyle geliştirilen yeni yöntem FedCBA olarak adlandırılmıştır. FedCBA, FP-Growth destekli bir CBA yaklaşımı olarak, federe öğrenme ortamında daha yüksek doğruluk ve verimlilik sağlamayı hedeflemektedir.

FedCBA yönteminin performansı, literatürde yaygın olarak kullanılan federe öğrenme birleştirme algoritmaları ile karşılaştırılmıştır. IBM İnsan Kaynakları Analitiği veri seti üzerinde daha önce gerçekleştirilmiş çalışmalarda kullanılan FedAvg algoritması referans alınmıştır [206]. Ayrıca, FedAvg algoritmasının heterojen (Non-IID) veri dağılımlarına yönelik geliştirilmiş bir versiyonu olan FedProx algoritması da karşılaştırmaya dahil edilmiştir. Literatürde FedAvg algoritması ile elde edilen sonuçlar mevcut çalışmadan alınırken, FedProx algoritması bu çalışma kapsamında aynı veri seti ve benzer deneysel koşullar altında uygulanarak değerlendirilmiştir.

Bu deneyde karşılaştırma amacıyla kullanılan FedAvg tabanlı çalışmada, IBM veri setinden eşit büyüklükte örneklemler oluşturulmuş, veri seti ADASYN yöntemi ile dengelenmiş ve temel bileşen analizi yöntemi uygulanarak boyut indirgeme işlemi gerçekleştirilmiştir. Veri dağılımı istemciler arasında kısmen IID olacak şekilde düzenlenmiştir. Söz konusu ön işleme adımları, kontrollü ve dengeli bir deney ortamı oluşturmak amacıyla uygulanmıştır. Önerilen FedCBA yaklaşımında ise dengeli veri seti istemciler arasında IID veri dağılımına uygun şekilde bölünmüş ve federe öğrenme süreci gerçekleştirilmiştir. Bu kapsamda elde edilen sonuçlar, literatürde yer alan FedAvg tabanlı yöntemler ile karşılaştırmalı olarak değerlendirilmiştir. Tablo 4.8.'de, kullanılan federe birleştirme algoritmaları, istemci tarafında tercih edilen model eğitim yöntemleri ve elde edilen doğruluk sonuçları sunulmaktadır.

Tablo 4.8. Dengelenmiş ve kısmen IID veri dağılımında federe birleştirme algoritmalarının performans karşılaştırması.

FL Birleştirme Algoritması	İstemcide Kullanılan Algoritma	Doğruluk (%)
FedAvg	Logistic Regression	87.00
FedAvg	Neural Network	89.00
FedCBA	CBA	87.00

FedAvg tabanlı yöntemler ile gerçekleştirilen karşılaştırmaların ardından, heterojen veri dağılımlarına yönelik geliştirilmiş bir yaklaşım olan FedProx algoritması ile de performans karşılaştırması gerçekleştirilmiştir. Bu kapsamda veri seti, istemciler arasında dengesiz ve Non-IID veri dağılımını temsil edecek şekilde bölünerek gerçek dünya federe öğrenme senaryoları simüle edilmiştir. Elde edilen sonuçlar, ilişkisel sınıflandırma tabanlı FedCBA yaklaşımının heterojen veri dağılımlarındaki performansını, gradient tabanlı FedProx yöntemi ile karşılaştırmalı olarak ortaya koymaktadır. Tablo 4.9.'da kullanılan federe birleştirme algoritmaları, istemci tarafında tercih edilen model eğitim yöntemleri ve doğruluk sonuçları sunulmaktadır.

Tablo 4.9. Dengesiz ve Non-IID veri dağılımında federe birleştirme algoritmalarının

performans karşılaştırması.

FL Birleştirme Algoritması	İstemcide Kullanılan Algoritma	Doğruluk (%)
FedProx	Multi Layer Perceptron	81.97
FedCBA	CBA	84.00

Tablo 4.8. incelendiğinde, FedCBA yaklaşımının doğruluk açısından literatürde yer alan FedAvg tabanlı yöntemlere yakın performans sergilediği görülmektedir. Tablo 4.8. incelendiğinde ise FedProx algoritması heterojen veri dağılımlarına yönelik geliştirilmiş olmasına rağmen, FedCBA'nın gerisinde kalmıştır. Bu durum, ilişkisel sınıflandırma temelli FedCBA yaklaşımının, rekabetçi bir performans sunduğunu göstermektedir.

4.4. FedFORGET Unutma Başarımı ve Sonuçları

Bu bölümde veriden habersiz bir şekilde hedeflenen verilerin modelden güvenli ve doğru biçimde unutulmasını sağlayan FedFORGET federe unutma yönteminin performansı değerlendirilmektedir. İstemcilerden gelen yerel modellerin sunucu tarafında birleştirilerek global modelin oluşturulması işlemi FedCBA tarafından sağlanırken, hedeflenen verilerin etkisinin bu birleştirilmiş global modelden unutulması görevi FedFORGET tarafından yürütülmektedir. Bu işleyiş doğrultusunda FedFORGET algoritması, ilişkisel sınıflandırma tabanlı FedCBA federe birleştirme algoritması ile doğrudan entegre edilerek işletilmiştir. Bu bölümde deney ortamının donanımsal ve yazılımsal kurulumu detaylandırılmakta, kullanılan veri setleri ve bu veri setleri üzerinden kurgulanan FU senaryoları açıklanmaktadır. Elde edilen bulgular model performans metrikleri ve unutma süresi üzerinden literatürdeki mevcut standart yaklaşım olan sıfırdan yeniden eğitim yöntemi ile karşılaştırmalı olarak sunulmaktadır.

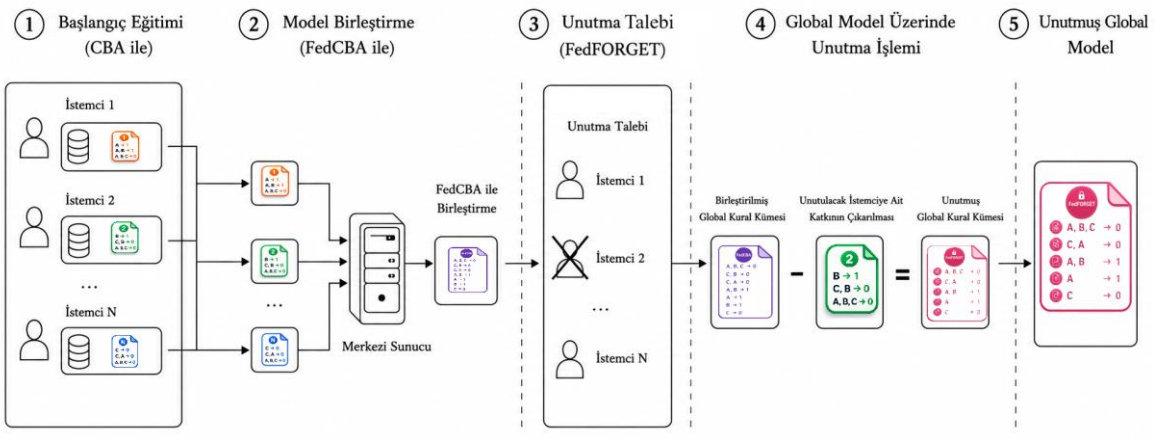
4.4.1. Deney ortamı

FedFORGET süreçlerine ait deneysel analizler, FedCBA için geliştirilen deney ortamı ile aynı olacak şekilde 11. Nesil Intel Core i5-1135G7 işlemciye ve 16 GB DDR3 belleğe sahip bir sistemde gerçekleştirilmiştir. FL ve MU ortamı, merkezi sunucu ile birden fazla istemci arasındaki iletişimi, model birleştirme ve birleştirilmiş model üzerinde unutma süreçlerini koordine etmek amacıyla simülasyon ortamı oluşturulmuştur. FedCBA için kurgulanan simülasyon ortamına FedFORGET mekanizmasının entegre edilmesiyle unutma aşaması yürütülmüştür. İstemci bazlı unutma için geliştirilmiş bu yöntemde, unutulacak istemci belirlendikten sonra ilgili istemciye ait bilgiler, veriye erişmeden doğrudan global model üzerinde formüller aracılığıyla unutturulmaktadır. Simülasyon ortamı FedFORGET'ın doğruluğunu değerlendirmek için kurulmuş olup, ağ gecikmesi gibi federe mimarinin

dezavantajlarını izole ederek kontrollü bir test ortamı sunmaktadır. FedFORGET algoritmasının mantıksal başarımını doğrulamaya odaklanılmıştır.

Sistemde, istemcilerin yerel modellerini eğitip sunucuda birleştirildiği süreçte FedCBA algoritması temel alınmıştır. CBA mimarisi kapsamında FP-Growth tabanlı ilişkisel kuralların çıkarımı için pyArc modülü, veri işleme, yapılandırma ve metrik hesaplamaları için ise NumPy ve pandas kütüphanelerinden yararlanılmıştır. Federe mimari simülasyonunda eğitim verisinin istemciler arasında bölünmesi yerel veri boyutlarını daralttığından, yüksek eşik değerlerinin kullanılması çok kısıtlı sayıda kural üretilmesine neden olmaktadır. Bu kısıtı aşmak, yerel modellerin yeterli büyüklükte bir kural havuzuna sahip olmasını sağlamak ve daraltılmış veri uzayındaki azınlık örüntülerini de yakalayabilmek amacıyla destek ve güven eşik değerleri 0.1 gibi düşük bir seviyede yapılandırılmıştır. Böylece azınlık örüntüler ile oluşan nadir kuralları da kapsayacak geniş bir kural seti elde edilmiştir.

Simülasyon ortamında verimliliği artırmak ve unutma sürecinin etkilerini daha net izole edebilmek amacıyla çok turlu iletişim yerine tek turlu bir yaklaşım benimsenmiştir. Bu yapılandırmaların tamamlanmasıyla birlikte, sistem hem başlangıç eğitimi hem de unutma senaryolarını test etmek üzere hazır hale getirilmiştir. İstemcilerin yerel modellerini eğitip sunucuya ilettiği, sunucuda FedCBA algoritması ile birleştirildiği ve FedFORGET ile unutma talebi ardından unutma işleminin gerçekleştiği içerik Şekil 4.8.'de görsel olarak sunulmaktadır.



Şekil 4.8. FedCBA ile federe birleştirme ve FedFORGET ile istemci bazlı unutma süreci.

Önerilen FedFORGET unutma algoritmasının performansını, genellenebilirliğini ve kararlılığını kapsamlı bir şekilde değerlendirebilmek amacıyla farklı öznitelik ve örneklem yapılarına sahip beş farklı veri seti kullanılmıştır. Bu veri setleri; IBM İnsan Kaynakları Analitiği, Adult, Mushroom, Cars Evaluation ve Nursery veri setleridir. Analizlerin

güvenilirliği için tüm veri setleri ilk aşamada %80 eğitim ve %20 test verisi olacak şekilde ayrılmıştır. Eğitim verisi, federe öğrenme mimarisini simüle etmek üzere belirli kurallara göre alt kümelere bölünmüştür. Her bir alt küme istemci olarak düşünülmüştür. Dağıtım sürecinde, yöntemin farklı koşullardaki esnekliğini ölçmek için hem IID hem de Non-IID senaryoları kurgulanmıştır.

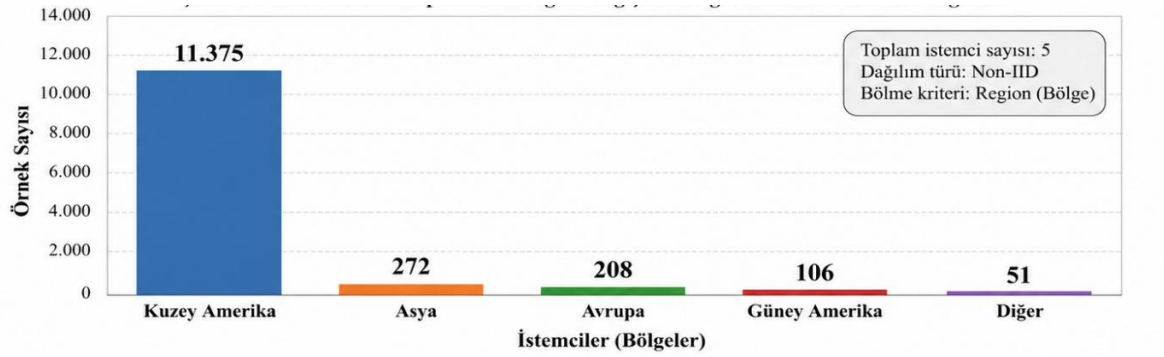
Non-IID dağılımda, gerçek dünya senaryolarını yansıtmak amacıyla veri, hedef etiketlere göre değil, öznitelik tabanlı anlamsal gruplara göre bölünerek dağıtılmış ve istemciler arası bir veri dengesizliği oluşturulmuştur. Bu kapsamda iki veri seti Non-IID olarak yapılandırılmıştır:

IBM İnsan Kaynakları Analitiği Veri Seti: Departman özniteliğine göre 3 istemciye bölünmüştür. Kurulan federe mimari simülasyonu kapsamında veri dağılımına ilişkin temel yapılandırma detayları Tablo 4.6.'da sunulmuş olup, istemcilere atanan veri dağılımı Şekil 4.6.'da görselleştirilmiştir.

Adult Veri Seti: Coğrafi konum bilgisine göre Güney Amerika, Avrupa, Asya, Kuzey Amerika, Diğerleri olarak adlandırılan 5 farklı bölgeye ayrılmıştır. Bu veri seti için kurulan federe mimari simülasyonu kapsamında veri dağılımına ilişkin temel yapılandırma detayları Tablo 4.10.'da sunulmuş olup, istemcilere atanan veri dağılımı Şekil 4.9.'da görselleştirilmiştir.

Tablo 4.10. Adult veri setinin federe mimari kapsamındaki veri dağılım parametreleri.

Özellik	Değerler
Bölme Kriteri	Bölge
İstemci Sayısı	5
İstemci İsimleri	Güney Amerika, Avrupa, Asya, Kuzey Amerika, Diğerleri
Veri Dağılım Türü (IID / Non-IID)	Non-IID



Şekil 4.9. Adult veri setinin federe öğrenme mimarisindeki istemcilere bölge bilgisine göre dağılımı.

FedFORGET deneylerinde kullanılan Adult veri setinde orijinal sınıf dağılımındaki dengesizliği gidermek ve modelin taraflı öğrenmesini engellemek amacıyla Random Undersampling [207] yöntemi uygulanmıştır. Orijinalinde 48.842 kayıt bulunan veri setindeki çoğunluk sınıfı, rastgele seçilerek azınlık sınıfının seviyesine indirilmiştir. Böylece her iki sınıf için eşit sayıda örneklem barındıran toplam 12.012 kayıtlık tamamen dengeli bir veri seti elde edilmiştir. Bu veri setine ait temel özellikler Tablo 4.2.'de sunulmaktadır.

IID dağılım senaryosunda veri, istemciler arasında rastgele ve eşit miktarlarda paylaştırılmıştır. Bu senaryo kapsamında Mushroom, Cars Evaluation ve Nursery veri setleri kullanılmıştır. Deneysel değerlendirmelerde adil ve tutarlı bir kıyaslama zemini oluşturabilmek amacıyla, kullanılan bu IID veri setlerinin tamamı standart olarak 5 istemciye bölünecek şekilde yapılandırılmıştır. Ancak istemcilere atanan toplam veri sayıları eşit olsa da, dağıtımın rastgele yapılması ve veri setlerinin kendi içlerindeki dengeli ya da dengesiz sınıf yapıları sebebiyle her istemcideki etiket dağılımı birebir aynı değildir. Ayrıca, algoritmanın farklı problem tiplerindeki başarısını ve esnekliğini ölçmek adına bu veri setleri hem ikili hem de çok sınıflı yapıları içerecek şekilde seçilmiştir. Çalışmada kullanılan beş veri setine ait istemci dağılımları ve temel bölme stratejileri Tablo 4.10'da özetlenmektedir.

Tablo 4.11. Federe öğrenme ve unutma süreçleri için veri seti dağılım stratejileri.

Veri Seti	İstemci Sayısı	Dağılım Türü	Bölme Mantığı
IBM	3	Non-IID	Departman Bazlı
Adult	5	Non-IID	Coğrafi Bölge Bazlı
Mushroom	5	IID	Rastgele Eşit Dağılım
Cars Evaluation	5	IID	Rastgele Eşit Dağılım
Nursery	5	IID	Rastgele Eşit Dağılım

4.4.2. Deneysel bulgular ve performans karşılaştırması

Bu çalışmada, önerilen unlearning yönteminin etkinliğini ölçmek amacıyla veri setleri gerçek dünya senaryolarına uygun şekilde istemci yapıları halinde bölünmüştür. Çalışmada kullanılan veri setleri şu şekildedir: IBM veri seti departmanlara göre 3 istemciye ayrılmış ve Sales departmanı unutulacak hedef istemci olarak belirlenmiştir. Adult veri seti bölgelere göre 5 istemciye ayrılmış ve Europe bölgesi unutulacak hedef istemci olarak seçilmiştir. Mushroom veri seti IID olarak 5 istemciye bölünmüş ve 5 numaralı istemci unutulacak hedef istemci olarak belirlenmiştir. Cars veri seti IID olarak 5 istemciye bölünmüş ve 5

numaralı istemci unutulacak hedef istemci olarak seçilmiştir. Nursery veri seti ise IID olarak 5 istemciye bölünmüş ve 5 numaralı istemci unutulacak hedef istemci olarak belirlenmiştir.

Analiz aşamasında performans karşılaştırması için kullanılan temel yöntemler aşağıda verilmiştir:

• Federe Eğitim (Federated Training - FedCBA): Bu çalışmada FL süreci, simülasyon ortamında gerçekleştirilmiştir. Eğitim verisi, belirlenen istemci sayısı kadar alt parçaya bölünmüş ve her bir parça bağımsız bir istemciyi temsil edecek şekilde yerel olarak işlenmiştir. İstemcilerde eğitilen yerel CBA modeller FedCBA birleştirme algoritması kullanılarak bir araya getirilmiş ve global model oluşturulmuştur.

• Federe Unutma (Federated Unlearning - FedFORGET): Bu çalışma kapsamında önerilen FedFORGET yöntemidir. Simülasyon ortamında belirlenen unutulacak hedef istemciye ait model katkısının, tüm sistemi baştan eğitmeye gerek kalmadan global modelden çıkarılmasını amaçlayan unutma sürecidir. Bu süreçte yalnızca hedef istemcinin modele sağladığı katkı kaldırılmakta ve güncellenmiş unutmuş global model elde edilmektedir.

• Federe Yeniden Eğitim (Federated Retraining - FedRetrain): Bu yöntemde unutulacak hedef istemciye ait veri, eğitim veri setinden tamamen çıkarılmakta ve kalan istemci verileri kullanılarak model simülasyon ortamındaki federe mimari üzerinde baştan yeniden eğitilmektedir. Bu yaklaşım, önerilen FedFORGET yönteminin doğruluğunu ve unutma başarısını değerlendirmek amacıyla referans yöntem olarak kullanılmaktadır.

Bu çalışmada kullanılan veri setleri, modelin öğrenme ve unutma davranışını analiz edebilmek amacıyla test verisi ve hedef veri olmak üzere iki temel gruba ayrılmaktadır. Test verisi, modelin genel performansını değerlendirmek için kullanılan ve eğitim sürecine dahil edilmeyen veri setini ifade etmektedir. Bu veri seti üzerinden elde edilen sonuçlar, modelin genelleme yeteneğini ve farklı veri dağılımlarında ne kadar başarılı olduğunu göstermektedir. Bu çalışma kapsamında her veri seti %80 eğitim %20 test olarak ayrılmıştır.

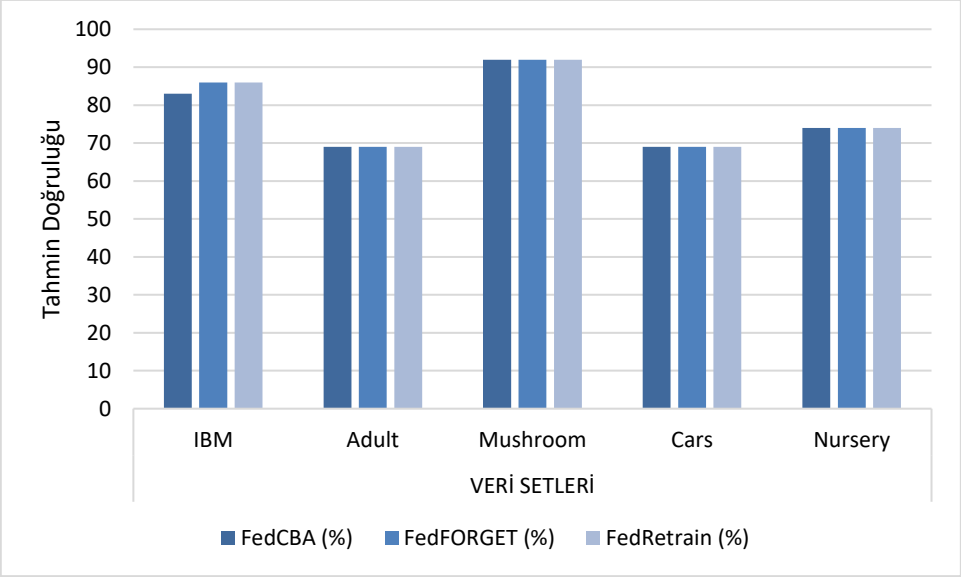
Hedef veri ise FU sürecinin temel bileşenini oluşturmaktadır. Bu veri, modelden çıkarılması istenen istemciye ait örnekleri temsil etmektedir. Başka bir ifadeyle, modelin artık hatırlamaması gereken bilgi, hedef veri kapsamında tanımlanmaktadır. Bu nedenle unutma işlemi sonrasında modelin hedef veri üzerindeki performansında düşüş gözlemlenmesi beklenmektedir. Bu düşüş unutma işleminin başarılı bir şekilde gerçekleştiğinin göstergesidir.

Tablo 4.12.'de, eğitim ve unutma işlemleri sonrasında test verileri üzerinde elde edilen model başarısı dört temel metrik üzerinden ayrıntılı olarak sunulmaktadır. Sonuçlar için kullanılan metrikler; genel doğru tahmin oranını ifade eden Doğruluk, modelin pozitif tahminlerinin doğruluk oranını ifade eden Kesinlik, gerçek pozitif örneklerin ne kadarının doğru yakalandığını gösteren Duyarlılık ve kesinlik ile duyarlılık arasındaki dengeli performansı temsil eden F1-Skorudur.

Tablo 4.12. Test verileri üzerinde elde edilen performans sonuçları.

Veri Seti	Yöntem	Doğruluk (%)	Kesinlik (%)	Duyarlılık (%)	F1-Skor (%)
IBM	FedCBA	83	81	83	81
	FedFORGET	86	84	86	84
	FedRetrain	86	84	86	84
Adult	FedCBA	69	73	69	68
	FedFORGET	69	73	69	67
	FedRetrain	69	73	69	67
Mushroom	FedCBA	92	100	92	97
	FedFORGET	92	100	92	97
	FedRetrain	92	100	92	97
Cars	FedCBA	69	56	69	62
	FedFORGET	69	58	69	63
	FedRetrain	69	58	69	63
Nursery	FedCBA	74	72	74	72
	FedFORGET	74	72	74	72
	FedRetrain	74	72	74	72

Şekil 4.10.'da, modelin federe eğitim ve federe unutma süreçlerindeki test verileri ile tahmin doğruluğu sonucu genel başarısına ilişkin karşılaştırmalı bir analiz sunulmakta ve önerilen yöntemin referans yöntem olan FedRetrain yaklaşımına karşı etkinliği gösterilmektedir.



Şekil 4.10. Test verileri için tahmin doğruluğu.

Şekil 4.10.'a göre, FedFORGET işlemi sonrasında test verileri üzerinde elde edilen doğruluk değerlerinin, FedCBA ile gerçekleştirilen başlangıç eğitim sonuçlarına kıyasla korunması ve bazı veri setlerinde benzer ya da daha yüksek seviyelerde gerçekleşmesi, önerilen yöntemin modelin genel öğrenme kapasitesini sürdürebildiğini göstermektedir. Elde edilen sonuçlar, unutma işleminin modelin tamamını olumsuz etkilemek yerine yalnızca hedef istemciye ait bilginin model üzerindeki etkisini kaldırmaya odaklandığını ortaya koymaktadır. Ayrıca FedFORGET sonuçlarının FedRetrain yöntemi ile benzerlik göstermesi, önerilen yaklaşımın modeli baştan yeniden eğitmeye gerek kalmadan etkili bir unutma işlemi gerçekleştirebildiğini göstermektedir. Bu durum, modelin genel sınıflandırma başarısı korunurken yalnızca hedeflenen bilginin seçici olarak modelden çıkarılabildiğini doğrulamaktadır.

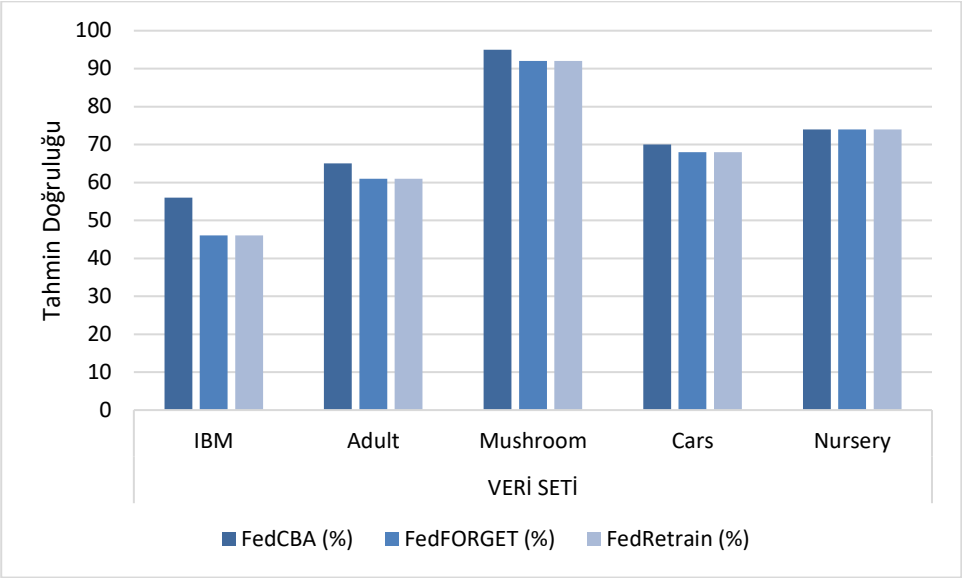
Tablo 4.13.'te, eğitim ve unutma süreçleri sonrasında unutulması hedeflenen istemci verileri üzerinde elde edilen performans sonuçları dört temel metrik üzerinden ayrıntılı olarak sunulmaktadır.

Tablo 4.13. Hedef veriler üzerinde elde edilen performans sonuçları.

Veri Seti	Yöntem	Doğruluk (%)	Kesinlik (%)	Duyarlılık (%)	F1-Skor (%)
IBM	FedCBA	56	71	56	62
	FedFORGET	46	69	46	55
	FedRetrain	46	69	46	55
Adult	FedCBA	65	72	65	64
	FedFORGET	61	77	61	68
	FedRetrain	61	77	61	68
Mushroom	FedCBA	95	100	95	97
	FedFORGET	92	100	92	95
	FedRetrain	92	100	92	95
Cars	FedCBA	70	56	70	62
	FedFORGET	68	58	68	63
	FedRetrain	68	58	68	63
Nursery	FedCBA	74	73	74	73
	FedFORGET	74	73	74	73
	FedRetrain	74	73	74	73

Tablo 4.13.'e göre, FedFORGET işlemi sonrasında hedef veriler üzerinde elde edilen metrik değerlerinin, eğitim aşamasında elde edilen sonuçlara kıyasla belirgin şekilde düştüğü görülmektedir. Bununla birlikte, tüm veri setlerinde FedFORGET sonuçlarının FedRetrain yöntemi ile örtüşmesi, önerilen yaklaşımın unutulması hedeflenen bilginin model üzerindeki etkisini başarıyla ortadan kaldırdığını göstermektedir. Elde edilen bu sonuçlar, FedFORGET yönteminin modeli baştan yeniden eğitmeye gerek kalmadan teorik olarak unutma başarısını sağlayabildiğini göstermektedir.

Şekil 4.11.'de, unutulması hedeflenen istemci verileri üzerinde global model ile unutma işlemi uygulanmış modellerin performans sonuçları gösterilmektedir.



Şekil 4.11. Hedef veriler için tahmin doğruluğu.

Şekil 4.10. ve Şekil 4.11. birlikte değerlendirildiğinde, test verileri üzerinde gerçekleştirilen deneylerde model performansının büyük ölçüde korunduğu görülürken; hedef veriler üzerinde gerçekleştirilen deneylerde performansın belirgin şekilde azalması, önerilen algoritmanın yalnızca unutulması hedeflenen bilgiyi modelden çıkardığını doğrulamaktadır. Bu durum, FedFORGET yaklaşımının modelin genel öğrenme kapasitesini korurken hedef istemciye ait bilgiyi seçici olarak temizleyebildiğini ortaya koymaktadır.

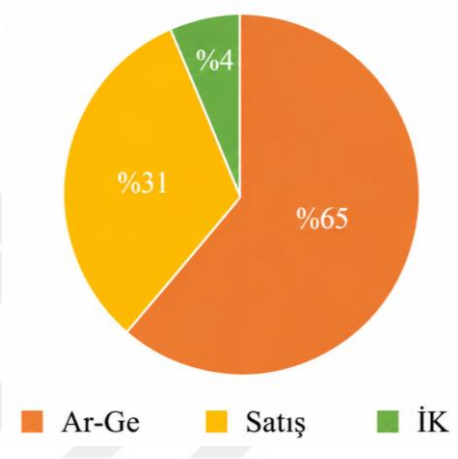
FL ortamlarında, unutma işleminin başarısı kadar işlem hızı da büyük önem taşımaktadır. Geleneksel yaklaşım olan FedRetrain yöntemi, unutulması istenen veri çıkarıldıktan sonra tüm modelin baştan yeniden eğitilmesini gerektirmekte, bu durum ise yüksek hesaplama maliyetine ve zaman kaybına neden olmaktadır. Buna karşılık FedFORGET yöntemi, tüm modeli yeniden eğitmek yerine yalnızca hedef istemciye ait kuralların güncellenmesini işleyerek süreci optimize etmeyi amaçlamaktadır. Tablo 4.14.'te yöntemlere ait model oluşturma ve unutma süreleri saniye cinsinden karşılaştırılmış ve bilimsel gösterim (10^n formatı) kullanılarak sunulmuştur.

Tablo 4.14. Federe öğrenme ve unutma sürelerinin karşılaştırılması (Saniye).

Veri Seti	FedCBA	FedFORGET	FedRetrain
IBM	7.97×10^{-2}	7.79×10^{-2}	7.79×10^{0}
Adult	2.31×10^{-2}	1.58×10^{-2}	3.88×10^{0}
Mushroom	1.40×10^{-2}	3.15×10^{-3}	2.01×10^{2}
Cars	1.04×10^{-3}	5.07×10^{-3}	6.89×10^{-2}
Nursery	1.37×10^{-3}	2.31×10^{-3}	6.32×10^{-1}

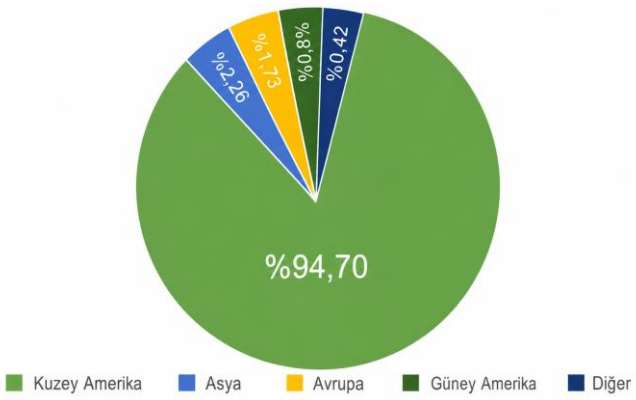
Çalışma kapsamında gerçekleştirilen deneysel senaryoda, her veri setinin eğitim verisi istemcileri temsil edecek şekilde stratejik olarak alt parçalara ayrılmıştır. Bu veri setlerinden IBM ve Adult veri setleri, gerçek dünya senaryolarını yansıtabilmek amacıyla özellikle Non-IID yapıda dağıtılmıştır. Bu başlık altında, Non-IID veri dağılımının etkilerini incelemek amacıyla her iki veri setinde de her istemci sırasıyla hedef istemci olarak seçilmiş ve FedFORGET algoritmasının performansı Doğruluk, Kesinlik, Duyarlılık ve F1-Skor metrikleri üzerinden ayrı ayrı değerlendirilmiştir.

Şekil 4.12.'de IBM veri setinin Sales, Ar-Ge ve İK departmanlarına göre dağılımı yer almaktadır. Veri dağılımı, gerçek dünya senaryolarına uygun olacak şekilde Non-IID ve dengesiz yapıdadır.



Şekil 4.12. IBM veri setinin departman bazlı istemci dağılımı.

Şekil 4.13.'de Adult veri setinin bölgelere göre dağılımı gösterilmekte olup, Kuzey Amerika (%94.7) baskın istemci konumundadır. Bu veri seti içinde veri dağılımı, gerçek dünya senaryolarına uygun olacak şekilde Non-IID ve dengesiz yapıdadır.



Şekil 4.13. Adult veri setinin bölge bazlı istemci dağılımı.

Tablo 4.15 ve Tablo 4.16.'da, Non-IID veri dağılımına sahip IBM ve Adult veri setlerinde

sırasıyla hedef istemci olarak seçilen istemciler için FedFORGET yöntemine ait performans sonuçları özetlenmektedir. Sonuçlar "Test" (unutulmayan genel veri) ve "Hedef" (unutulması amaçlanan istemci verisi) olmak üzere iki ayrı kategori altında sunulmuştur.

Tablo 4.15. Non-IID ortamda sırasıyla hedef istemciler unutulduktan sonra test verileri üzerinde elde edilen performans sonuçları.

Veri Seti (Hedef İstemci)	Doğruluk (%)	Kesinlik (%)	Duyarlılık (%)	F1-Skor (%)
IBM_Satış	86	84	86	84
IBM_ Ar-Ge	75	80	76	78
IBM_İK	66	82	66	71
Adult_Asya	69	73	69	68
Adult_Avrupa	69	73	69	67
Adult_Kuzey Amerika	68	72	68	66
Adult_Güney Amerika	74	81	75	77
Adult_Diğer	76	80	78	79

Tablo 4.16. Non-IID ortamda sırasıyla hedef istemciler unutulduktan sonra hedef veriler üzerinde elde edilen performans sonuçları.

Veri Seti (Hedef İstemci)	Doğruluk (%)	Kesinlik (%)	Duyarlılık (%)	F1-Skor (%)
IBM_Satış	46	69	46	55
IBM_ Ar - Ge	51	57	51	54
IBM_İK	61	63	68	62
Adult_Asya	65	74	66	70
Adult_Avrupa	61	77	61	68
Adult_Kuzey Amerika	67	71	67	66
Adult_Güney Amerika	72	73	72	73
Adult_Diğer	72	73	72	72

IBM veri seti sonuçları incelendiğinde, hedef istemci olarak Satış departmanının seçildiği senaryoda test verileri üzerindeki doğruluk değerinin %86 seviyesinde korunduğu görülmektedir. Buna karşın hedef veri üzerindeki doğruluk değerinin %46'ya düşmesi, FedFORGET yönteminin hedef istemciye ait bilgiyi modelden başarılı bir şekilde uzaklaştırabildiğini göstermektedir. Satış departmanının veri içerisindeki baskın yapısı göz önüne alındığında, genel model performansının korunmasına rağmen hedef veri üzerindeki başarının belirgin şekilde azalması, seçici unutma işleminin etkinliğini ortaya koymaktadır.

Ar-Ge departmanının hedef istemci olarak seçildiği senaryoda, test verileri üzerindeki doğruluk değeri %75 olarak elde edilirken hedef veri üzerindeki doğruluk değeri %51'e düşmüştür. Bu durum, modelin genel öğrenme kapasitesini büyük ölçüde koruduğunu ancak hedef istemciye ait örüntülerin önemli ölçüde temizlendiğini göstermektedir. Satış

senaryosuna kıyasla test performansındaki düşüş, Ar-Ge departmanına ait veri örüntülerinin global model üzerinde daha belirgin bir etkiye sahip olduğunu göstermektedir.

İK departmanının hedef istemci olduğu durumda ise test verileri üzerindeki doğruluk değeri %66, hedef veri üzerindeki doğruluk değeri ise %61 olarak elde edilmiştir. İK istemcisinin veri içerisindeki oranının düşük olması nedeniyle modelin başlangıçta bu istemciyi daha sınırlı düzeyde öğrendiği değerlendirilmektedir. Bu nedenle unutma sonrasındaki performans kaybı diğer istemcilere kıyasla daha düşük seviyede gerçekleşmiştir.

Adult veri seti sonuçları incelendiğinde, Asya bölgesinin hedef istemci olarak seçildiği senaryoda test verileri üzerindeki doğruluk değeri %69 seviyesinde korunurken hedef veri üzerindeki doğruluk değeri %65'e düşmüştür. Sonuçlar, modelin genel bilgisini koruyabildiğini ancak hedef bölgeye ait örüntüler üzerinde belirli düzeyde unutma gerçekleştirdiğini göstermektedir.

Avrupa bölgesinin hedef istemci olarak seçildiği durumda test doğruluğu %69 iken hedef veri doğruluğu %61 olarak elde edilmiştir. Test ve hedef veri sonuçları arasındaki farkın daha belirgin olması, Avrupa bölgesine ait bazı örüntülerin modelden daha etkili şekilde temizlendiğini göstermektedir.

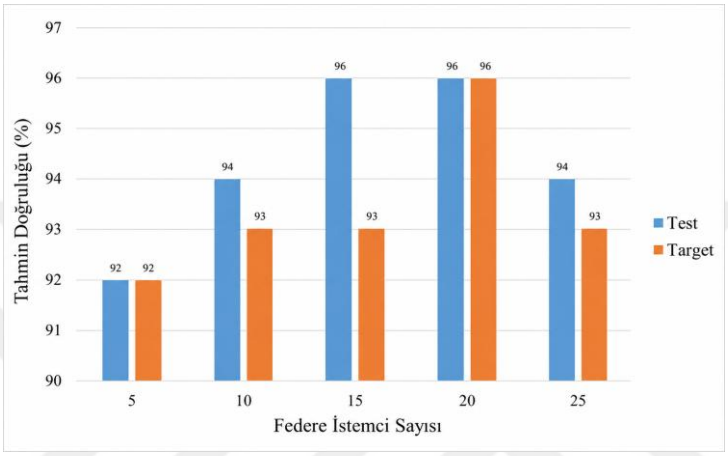
Kuzey Amerika istemcisinin hedef olarak seçildiği senaryoda, test doğruluğu %68, hedef veri doğruluğu ise %67 olarak gerçekleşmiştir. Veri setindeki baskın istemci olmasına rağmen unutma sonrasında performansın büyük ölçüde korunması, bölgeler arasında yüksek düzeyde anlamsal benzerlik bulunduğunu göstermektedir. Model, Kuzey Amerika verilerini unuttuktan sonra dahi diğer bölgelerden öğrenilen ortak kurallar aracılığıyla doğru tahminler gerçekleştirebilmiştir.

Güney Amerika istemcisinin hedef olarak seçildiği senaryoda test doğruluğu %74, hedef veri doğruluğu ise %72 olarak elde edilmiştir. Test ve hedef veri sonuçlarının birbirine oldukça yakın olması, ilgili bölgeye ait örüntülerin diğer istemciler tarafından da desteklendiğini ve modelin ortak kurallar üzerinden tahmin üretmeye devam ettiğini göstermektedir.

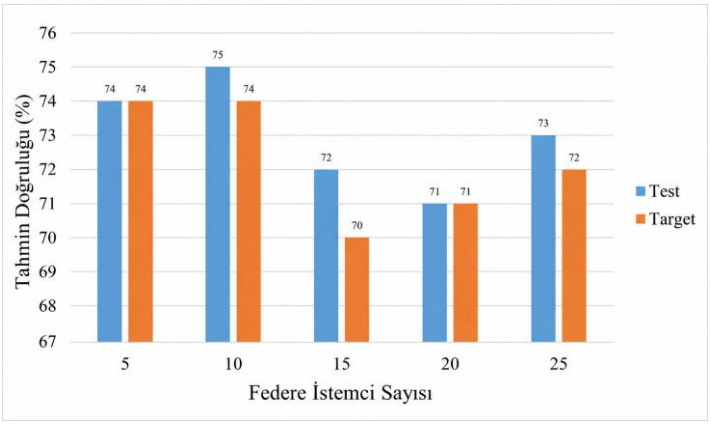
Diğer bölgeler istemcisinin hedef olarak seçildiği senaryoda ise test doğruluğu %76, hedef veri doğruluğu %72 olarak elde edilmiştir. Sonuçlar, FedFORGET yönteminin modelin genel performansını korurken hedef istemciye ait bilgiyi belirli ölçüde azaltabildiğini göstermektedir. Adult veri seti genel olarak değerlendirildiğinde, bölgeler arasında bulunan

ortak anlamsal örüntülerin modelin genelleme yeteneğini desteklediği ve unutma sonrasında performans kaybını sınırladığı görülmektedir.

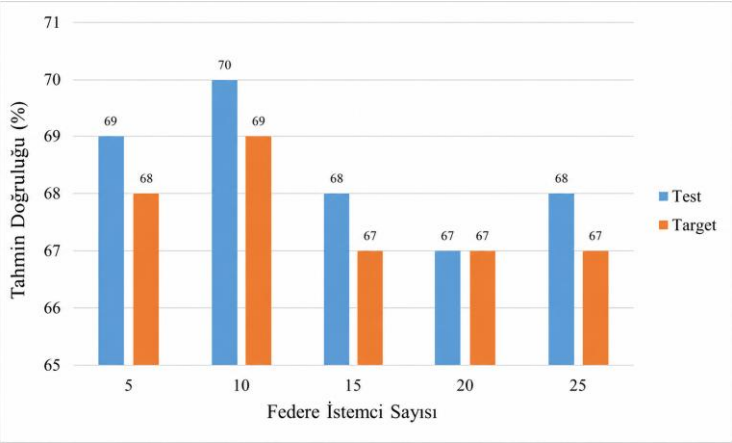
Çalışma kapsamında ayrıca FedFORGET yönteminin farklı ağ yoğunluklarındaki ölçeklenebilirliğini değerlendirebilmek amacıyla Mushroom, Nursery ve Cars Evaluation veri seti 5, 10, 15, 20 ve 25 istemci şeklinde ayrılmıştır. Her senaryoda unutma deneyleri gerçekleştirilmiş ve ağdaki istemci sayısı ile modelin genelleme yeteneği arasındaki ilişki incelenmiştir. Şekil 4.14.'te Mushroom veri seti için, Şekil 4.15.'te Nursery veri seti ve Şekil 4.16.'da Cars Evaluation veri seti için farklı istemci sayılarında FedFORGET yönteminin performans sonuçları yer almaktadır.



Şekil 4.14. Mushroom veri seti için farklı istemci sayılarında FedFORGET yönteminin performans sonuçları.



Şekil 4.15. Nursery veri seti için farklı istemci sayılarında FedFORGET yönteminin performans sonuçları.



Şekil 4.16. Cars Evaluation veri seti için farklı istemci sayılarında FedFORGET yönteminin performans sonuçları.

Literatürde yer alan derin öğrenme tabanlı federe unutma çalışmaları genellikle kara kutu modeller kullanılarak gerçekleştirilmekte olup, modelin iç karar mekanizmasına ilişkin şeffaflık sunmamaktadır. Buna karşın, önerilen ilişkisel kural tabanlı yaklaşım sayesinde modelin hangi bilgiyi öğrendiği, hangi bilgiyi unuttuğu ve bunun neden gerçekleştiği doğrudan gözlemlenebilmektedir.

Bu çalışmada eğitilen tüm modeller okunabilir .txt formatında kural listeleri şeklinde saklanmaktadır. Böylece herhangi bir uzman, modelin içerdiği öznitelikleri (örneğin; *safety*, *persons*, *buying*) ve karar verme mekanizmasını kolaylıkla yorumlayabilmektedir. Bu durum şeffaflık ve okunabilirlik açısından önemli bir avantaj sağlamakla birlikte, hassas bilgilerin korunması amacıyla model dosyalarının şifrelenmesi gibi ek güvenlik önlemleri de uygulanabilir. Ancak bu güvenlik önlemleri çalışmanın kapsamı dışında bırakılmıştır.

FL eğitim sürecinde istemcilerden elde edilen sınıflandırma kuralları, FedCBA birleştirme algoritması aracılığıyla destek ve güven değerleri temel alınarak global modele entegre edilmektedir. Deneysel çalışmada Cars Evaluation veri seti beş farklı istemciye dağıtılmış ve İstemci 3, MU için hedef istemci olarak belirlenmiştir. Tablo 4.17.'de, istemcilerin yerel CBA eğitimi sonucunda ürettiği kuralların dağılımı ve global modelin bu kuralları hangi istemcilerden devraldığı gösterilmektedir.

Tablo 4.17. Cars Evaluation veri seti kullanılarak global modelin oluşturulması ve istemci katkılarının analizi.

Kurallar	Kaynak İstemci(ler)	Destek	Güven
safety = low ⇒ unacc	İstemci 1, 2, 3, 4, 5	0.3364	1.0000
persons = 2 ⇒ unacc	İstemci 1, 2, 3, 4	0.3292	1.0000
buying = vhigh ⇒ unacc	İstemci 3, 5	0.0912	0.8750
maint = vhigh ⇒ unacc	İstemci 1, 2	0.0897	0.8733

lug_boot = small ⇒ unacc	İstemci 3	0.0550	0.8172

Unutma aşamasında FedFORGET, hedef istemciye (İstemci 3) ait etkileri kaldırmak amacıyla kural formüllerini işlemektedir. Bu süreçte yalnızca hedef istemciye özgü olan kurallar modelden tamamen çıkarılmaktadır. Hedef istemciden kaynaklanan ancak diğer istemcilerde de bulunan kurallar ise modelde tutulmaya devam etmektedir. Ancak bu kuralların destek ve güven değerleri, hedef istemcinin katkısı çıkarılarak yeniden güncellenmektedir. Tablo 4.18.'de FedFORGET işlemi sonrasında modelin kural yapısında meydana gelen değişimler gösterilmektedir.

Tablo 4.18. Cars Evaluation veri setinde global model oluşturulduktan sonra İstemci 3'ün unutulması sonrası kural ve performans metriklerindeki değişim.

Kurallar	**Unutma Öncesi (FedCBA)**	**Unutma Sonrası (FedFORGET)**	**Açıklama / Sonuç**
safety = low ⇒ unacc	Sup: 0.3364, Conf: 1.0000	Sup: 0.3334, Conf: 1.0000	İstemci 3 katkısı çıkarılarak güncellendi
persons = 2 ⇒ unacc	Sup: 0.3292, Conf: 1.0000	Sup: 0.3200, Conf: 1.0000	İstemci 3 katkısı çıkarılarak güncellendi
maint = vhigh ⇒ unacc	Sup: 0.0897, Conf: 0.8733	Sup: 0.0900, Conf: 0.8700	Kalan istemciler üzerinden güncellendi
buying = vhigh ⇒ unacc	Sup: 0.0912, Conf: 0.8750	Sup: 0.0700, Conf: 0.8500	İstemci 3 katkısı çıkarılarak güncellendi
lug_boot = small ⇒ unacc	Sup: 0.0550, Conf: 0.8172	SİLİNDİ	Hedef istemciye özgü kural kaldırıldı

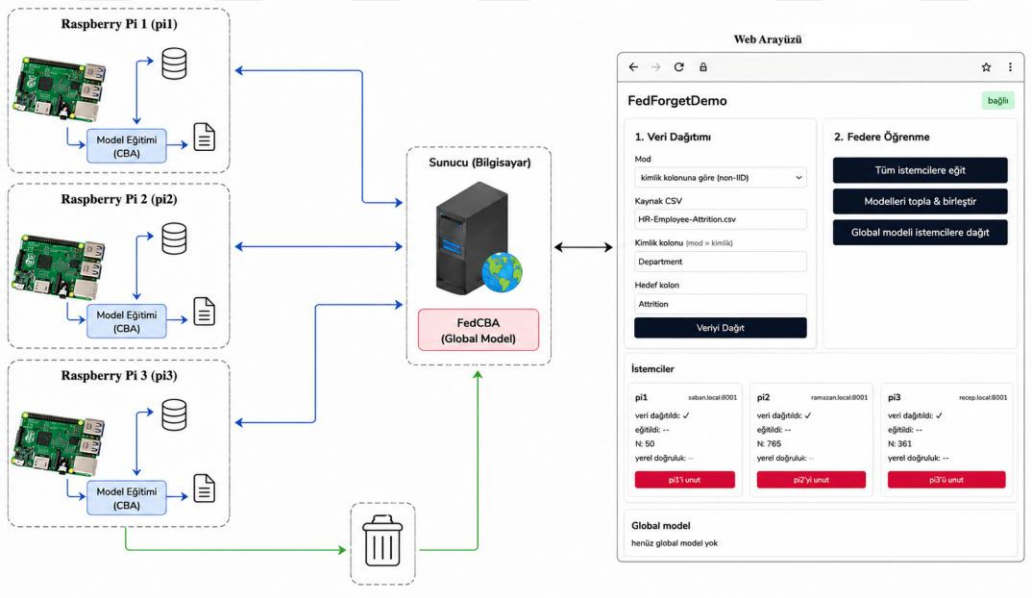
Tablo 4.18. üzerinden gerçekleştirilen analiz, FedFORGET mimarisinin kural seviyesinde seçici unutma yeteneğini somut verilerle doğrulamaktadır. Analiz sonucunda, hedef İstemci 3 ile diğer istemciler arasında ortak olarak bulunan genel kuralların modelden silinmediği, bunun yerine destek ve güven değerlerinin, hedef istemcinin katkısı çıkarılarak güncellendiği görülmektedir. Örneğin *safety = low ⇒ unacc*, *persons = 2 ⇒ unacc* ve *buying = vhigh ⇒ unacc* gibi kurallar, hedef istemci dışındaki istemciler tarafından da desteklendiği için model içerisinde korunmuştur. Bu durum, modelin genel faydasının korunabildiğini göstermektedir. Buna karşılık yalnızca hedef istemci tarafından üretilen kurallar global modelden tamamen kaldırılmıştır. Örneğin *lug_boot = small ⇒ unacc* kuralı yalnızca İstemci 3'e ait olduğu için silinmiştir. Bu sonuç, hedef istemcinin izlerinin, ortak bilgiyi gereksiz şekilde silmeden seçici olarak modelden çıkarılabildiğini göstermektedir. Tüm bu değişimlerin doğrudan metin dosyaları üzerinden takip edilebilmesi ise sistemin şeffaflığını ortaya koymaktadır. Böylece silme ve güncelleme işlemleri matematiksel metrikler üzerinden incelenebilmekte ve doğrulanabilmektedir.

4.5. Raspberry Pi İstemcileri ile Gerçekleştirilen Federe Unutma Performans Analizi

Bu bölümde, önerilen FedFORGET yaklaşımı Raspberry Pi cihazları üzerinde deneysel olarak test edilmiştir. Bu kurulumun temel amacı, yöntemin yalnızca yerel (local) bir ortamda değil, farklı fiziksel cihazlar üzerinde çalışan dağıtık istemcilerle de uygulanabilir olduğunu göstermektir. Bu kapsamda, her bir Raspberry Pi bağımsız bir istemci olarak konumlandırılmış ve yerel model eğitimi bu cihazlar üzerinde gerçekleştirilmiştir. Elde edilen sonuçlar, önerilen yöntemin merkezi olmayan ve farklı ortamlarda çalışan istemci yapıları altında da etkin bir şekilde çalışabildiğini ortaya koymaktadır.

4.5.1. Deney ortamı

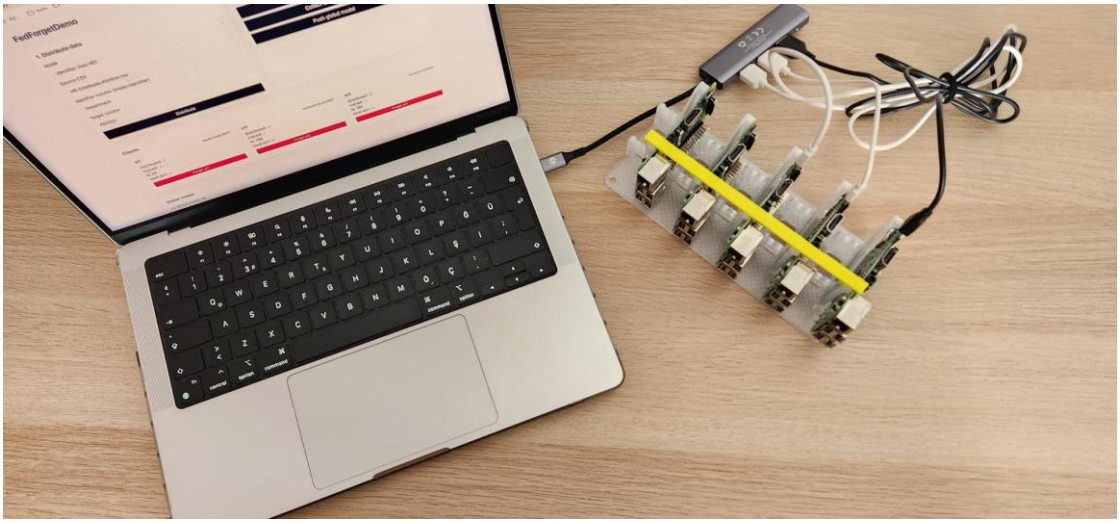
FedFORGET sistemi, üç adet Raspberry Pi cihazı ve bir sunucu bilgisayar kullanılarak gerçek ortamda uygulanmıştır. Her bir Raspberry Pi üzerinde yerel veri kullanılarak CBA tabanlı model eğitimi gerçekleştirilmiş, elde edilen yerel modeller birleştirilmek üzere sunucuya gönderilmiştir. Sunucu tarafında bu modeller FedCBA yöntemi ile birleştirilerek global model oluşturulmuştur. Daha sonra unutma talebinde bulunan istemci sistemden çıkarılmış ve global model, ilgili istemcinin katkısı kaldırılacak şekilde güncellenmiştir. Raspberry Pi tabanlı istemciler, merkezi sunucu ve localhost üzerinden çalışan yönetim arayüzünden oluşan sistem mimarisi Şekil 4.17.'de gösterilmektedir.



Şekil 4.17. Raspberry Pi tabanlı fiziksel istemciler, merkezi sunucu ve localhost arayüzünden oluşan FedFORGET sistem mimarisi.

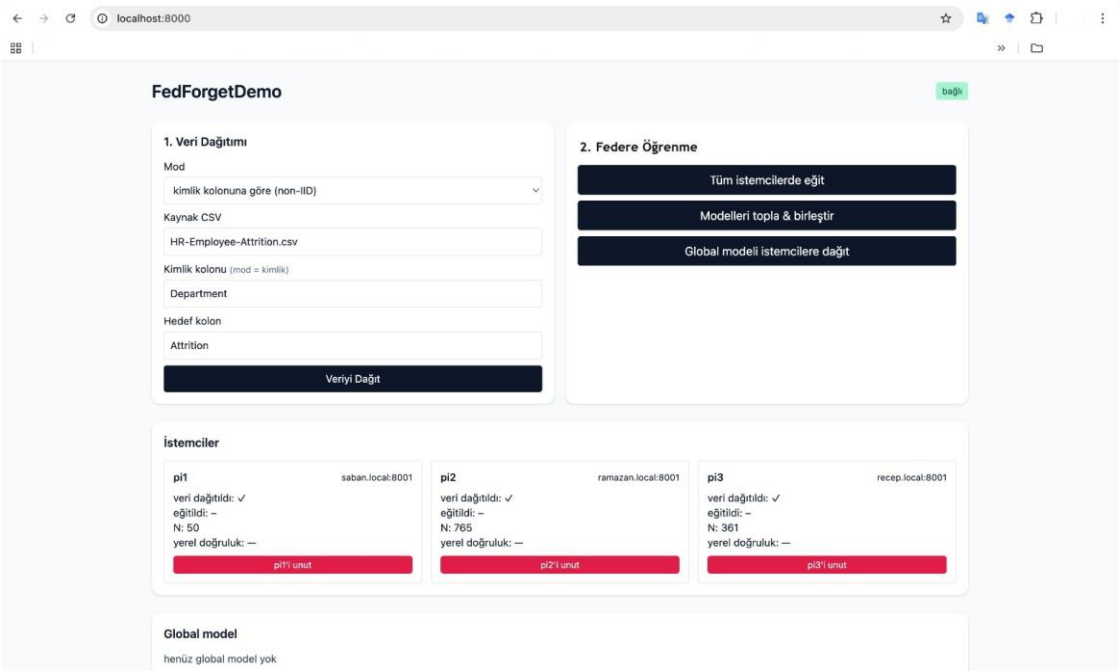
Deneysel kurulum sürecinde, istemci cihazı olarak Raspberry Pi 3B modelleri kullanılmıştır. Her bir cihaz için 64 GB kapasiteli SD kartlar tercih edilmiş ve cihazlara

Debian tabanlı Raspbian işletim sistemi kurulmuştur. Kurulum işlemlerinin ardından cihazlar aktif hale getirilerek dağıtık istemci yapısı oluşturulmuştur. Fiziksel çalışma düzeneği Şekil 4.18.'de sunulmaktadır.



Şekil 4.18. Deneysel çalışmalarda kullanılan Raspberry Pi tabanlı fiziksel istemci düzeneği.

FedFORGET yaklaşımının uygulanabilmesi amacıyla her bir Raspberry Pi üzerinde CBA algoritması çalıştırılarak yerel modeller eğitilmiştir. Sistem mimarisinde merkezi kontrolü sağlamak amacıyla Apple 14-Inch Macbook Pro M4 Chip 10 Core CPU And 10 Core GPU 24Gb 1TB SSD özelliklerine sahip bilgisayar sunucu olarak yapılandırılmıştır. Sunucu tarafında, yerel modellerin birleştirilmesi ve unutma işlemlerinin yönetilmesi için localhost üzerinde çalışan görsel bir arayüz geliştirilmiştir. Bu arayüz sayesinde sistemin durumu izlenebilmekte ve işlemler kullanıcı tarafından kontrol edilebilmektedir. Geliştirilen localhost tabanlı yönetim arayüzü Şekil 4.19.'da gösterilmektedir.



Şekil 4.19. Sunucu bilgisayar üzerinde çalışan localhost tabanlı yönetim ve kontrol arayüzü.

Bu çalışmada unutma işlemi kullanıcı arayüzü üzerinden manuel olarak tetiklenmektedir. Geliştirilen yapı, gerçek dünya senaryolarında istemcinin sistemden ayrılması veya güvenlik ihlali gibi durumlarda otomatik olarak çalışabilecek bir mekanizmanın simülasyonu niteliğindedir. Geliştirilen FedFORGETDemo arayüzü dört temel bileşenden oluşmaktadır. Veri Dağıtım Paneli, veri setinin istemcilere dağıtım şeklini belirlemek amacıyla kullanılmaktadır. Bu bölümde veri dağılım türü, kaynak veri seti, hedef değişken ve bölme kriteri gibi parametreler tanımlanabilmektedir.

Federe Öğrenme Paneli, federe öğrenme sürecinin temel adımlarını yönetmektedir. Bu panel üzerinden istemcilerde yerel model eğitimi başlatılabilmekte, modeller merkezi sunucuda birleştirilebilmekte ve oluşturulan global model yeniden istemcilere dağıtılabilmektedir.

Sistemde üç adet fiziksel istemci yer almakta olup, her bir istemciye ait bilgiler ayrı istemci kartları üzerinde gösterilmektedir. Görselde yer alan istemci panellerinde ilgili istemcinin veri dağıtım durumu, eğitim bilgileri, örnek sayısı ve yerel doğruluk değerleri sunulmaktadır. Süreç içerisinde tamamlanan işlemler arayüz üzerinde tik işareti ile belirtilerek sistem durumunun görsel olarak takip edilmesi sağlanmaktadır. Ayrıca unutma işleminin gerçekleştirilmesi amacıyla ilgili istemci panelinde bulunan unut butonu kullanılmakta ve FedFORGET süreci bu buton aracılığıyla tetiklenmektedir.

Global Model Paneli ise oluşturulan global modelin mevcut durumunu göstermektedir. Bu yapı sayesinde veri dağıtımı, model eğitimi, model birleştirme ve istemci bazlı unutma işlemleri bütünleşik bir şekilde yönetilebilmektedir.

Arayüz içerisinde, unutma işlemi sonrasında elde edilen performans sonuçlarını analiz etmeye yönelik bir değerlendirme bölümü de bulunmaktadır. Bu bölümde, global modelin birleştirme sonrası test performansı, unutma işlemi sonrasında test verisi üzerindeki performansı ve unutulan istemciye ait hedef veri üzerindeki performansı karşılaştırmalı olarak gösterilmektedir. Böylece unutma işleminin model üzerindeki etkisi hem sayısal değerler hem de grafiksel gösterimler aracılığıyla doğrudan gözlemlenebilmektedir.

Ayrıca arayüz içerisinde gerçekleştirilen tüm işlemleri anlık olarak izleyebilmeyi sağlayan bir günlük (log) mekanizması bulunmaktadır. Bu yapı sayesinde veri dağıtımı, yerel model

eğitimi, model birleştirme ve global model dağıtımı gibi süreçler kayıt altına alınmaktadır. Sistem süreçleri anlık olarak takip edilebilmektedir.

Bu çalışma kapsamında kullanılan veri yapıları, test veri ve hedef veri olmak üzere iki temel gruba ayrılmıştır. Test veri, modelin genel performansını değerlendirmek amacıyla kullanılan ve eğitim sürecine dahil edilmeyen veri kümesini ifade etmektedir. Hedef veri ise modelden çıkarılması istenen istemciye ait örnekleri temsil etmektedir. Bu nedenle unutma işlemi sonrasında hedef veri üzerindeki performansın düşmesi, unutma işleminin başarılı şekilde gerçekleştirildiğinin göstergesi olarak değerlendirilmektedir. Çalışma kapsamında veri seti %80 eğitim ve %20 test olacak şekilde ayrılmıştır.

4.5.2. Deneysel Bulgular

Bu çalışmada önerilen FedCBA ve FedFORGET mimarisinin performansı, IBM veri seti kullanılarak farklı veri dağılımı senaryoları altında değerlendirilmiştir. Bu bölümde elde edilen sonuçlar, önceki kontrollü simülasyon deneylerinden farklı fiziksel ve ağ koşulları altında gerçekleştirilmiştir. Bu nedenle performans metriklerinde küçük farklılıklar oluşabilmektedir.

Deneysel senaryoların özeti Tablo 4.19.'da sunulmaktadır. Deneyler, modelin hem öğrenme hem de unutma davranışını analiz edebilmek amacıyla IID ve Non-IID veri dağılımları üzerinde gerçekleştirilmiştir. İlk senaryoda veri seti departman bilgisine göre istemcilere ayrılmış ve böylece gerçek dünyadaki heterojen veri dağılımını temsil eden Non-IID yapı oluşturulmuştur. Bu senaryoda unutma işlemi hedef istemci olarak belirlenen Ar-Ge istemcisi üzerinde gerçekleştirilmiştir. İkinci senaryoda ise veri seti istemciler arasında rastgele ve dengeli biçimde dağıtılarak IID yapı oluşturulmuştur. IID senaryosunda ise unutma işlemi 2 numaralı istemci üzerinde uygulanmıştır.

Tablo 4.19. Raspberry Pi ortamında gerçekleştirilen deneysel senaryoların özeti.

Senaryo	Veri Dağılımı	İstemci Sayısı	Unutulacak Hedef İstemci	Açıklama
Senaryo 1	Non-IID	3 (Ar-Ge, İK, Satış)	Ar-Ge	Veri seti departmanlara göre bölünmüştür.
Senaryo 2	IID	3 (İstemci 1, İstemci 2, İstemci 3)	İstemci 2	Veri seti istemciler arasında rastgele ve dengeli biçimde bölünmüştür.

Deneysel sonuçlar, modelin öğrenme ve unutma performansını değerlendirebilmek

amacıyla farklı ölçümler üzerinden analiz edilmiştir. FedCBA Test ölçümü, federe öğrenme sonrasında oluşturulan global modelin test verisi üzerindeki doğruluğunu göstermektedir. FedFORGET Hedef ölçümü, unutulması talep edilen istemciye ait hedef veri üzerindeki doğruluk sonucunu ifade etmektedir. FedFORGET Test ölçümü ise unutma işlemi sonrasında modelin genel test verisi üzerindeki doğruluğunu göstermektedir. Bu ölçümler sayesinde modelin hem öğrenme başarısı hem de unutma işleminin etkinliği birlikte değerlendirilebilmektedir.

Tablo 4.20.'de Raspberry Pi tabanlı fiziksel istemci ortamında IID ve Non-IID veri dağılımları altında istemcilerde eğitilen yerel CBA modellerine ait doğruluk sonuçları sunulmaktadır. Non-IID senaryosunda veri seti departman bazlı olarak bölünmüş olup istemciler arasında belirgin performans farklılıkları oluştuğu görülmektedir. Özellikle Ar-Ge istemcisi en yüksek doğruluk değerine ulaşırken, Satış istemcisinin daha düşük performans gösterdiği gözlemlenmiştir. Bu durum, heterojen veri dağılımının istemci modelleri üzerinde doğrudan etkili olduğunu göstermektedir. IID senaryosunda ise veri setinin istemcilere daha dengeli dağıtılması nedeniyle doğruluk sonuçlarının birbirine daha yakın olduğu görülmektedir. Elde edilen sonuçlar, veri dağılım yapısının yerel model performansları üzerinde önemli etkisi bulunduğunu ortaya koymaktadır.

Tablo 4.20. Raspberry Pi ortamında IID ve Non-IID veri dağılımlarında yerel CBA modellerinin doğruluk sonuçları.

Veri Dağılımı	İstemci	CBA Test
Non-IID	İK	0,7619
	Ar-Ge	0,8571
	Satış	0,7380
IID	İstemci 1	0,8367
	İstemci 2	0,8231
	İstemci 3	0,7891

Tablo 4.21.'de Raspberry Pi tabanlı fiziksel istemci ortamında gerçekleştirilen FedCBA birleştirme ve FedFORGET unutma işlemlerine ait doğruluk sonuçları sunulmaktadır. Sonuçlar incelendiğinde, FedCBA yöntemi ile oluşturulan global modelin hem IID hem de Non-IID veri dağılımlarında yüksek doğruluk değerleri sağladığı görülmektedir. Non-IID senaryosunda veri heterojenliğine rağmen global model doğruluğunun korunabilmesi, önerilen FedCBA yaklaşımının farklı veri dağılımlarına karşı kararlı çalışabildiğini göstermektedir. Ayrıca FedFORGET işlemi sonrasında model performansında yalnızca sınırlı düzeyde düşüş meydana geldiği görülmektedir. Özellikle hedef veri üzerindeki doğruluk değişimi, unutma işleminin etkin şekilde gerçekleştirildiğini göstermektedir. Test verisi üzerindeki sonuçlar, modelin genel performansını büyük ölçüde koruyabildiğini

ortaya koymaktadır. Bu sonuçlar, önerilen yaklaşımın fiziksel dağıtık istemci ortamlarında da etkin şekilde uygulanabildiğini göstermektedir.

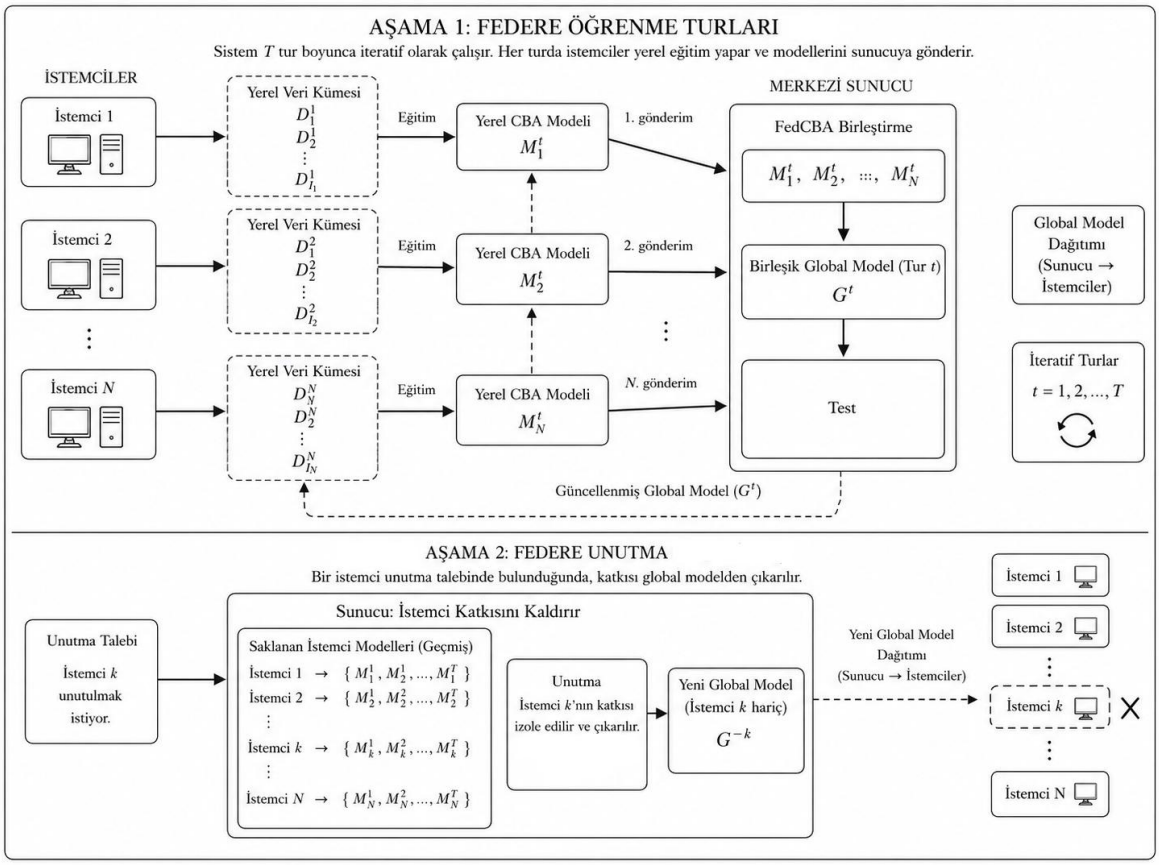
Tablo 4.21. Raspberry Pi ortamında FedCBA ve FedFORGET doğruluk sonuçları.

Veri Dağılımı	Unutulan İstemci	FedCBA Test	FedFORGET Hedef	FedFORGET Test
Non-IID	Ar-Ge	0,8639	0,8444	0,8435
IID	İstemci 2	0,8571	0,8265	0,8382

4.6. Bulut Ortamında Federe Unutma Performans Analizi

Bu bölümde, önerilen FedCBA ve FedFORGET mimarisinin bulut tabanlı dağıtık ortam üzerindeki performansı analiz edilmektedir. Çalışma kapsamında sistem yalnızca simülasyon ortamında değerlendirilmemiştir. İstemciler, API katmanı ve merkezi sunucudan oluşan istemci-sunucu mimarisi üzerinde uygulanmıştır. Bu amaçla Google Cloud altyapısı kullanılmış ve FL ile FU süreçleri dağıtık haberleşme mekanizmaları üzerinden gerçekleştirilmiştir. Böylece önerilen yöntemin gerçek dünya koşullarındaki uygulanabilirliği, istemci-sunucu haberleşmesine uyumu, model güncelleme süreçlerindeki davranışı ve dağıtık ortam altında FU işlemlerini gerçekleştirme başarısı analiz edilmiştir. Ayrıca sistemin yalnızca model performansı değil, veri aktarımı, bellek yönetimi ve unutma işlemlerinin dağıtık mimari üzerindeki çalışma davranışı da değerlendirilmiştir.

Önerilen mimaride istemciler, kendi yerel veri kümeleri üzerinde CBA algoritması ile yerel modeller üretmektedir. Eğitilen yerel modeller API katmanı aracılığıyla merkezi sunucuya aktarılmakta ve merkezi sunucu tarafında FedCBA algoritması kullanılarak birleştirilmektedir. Birleştirme işlemi sonucunda oluşturulan global model tekrar istemcilere gönderilmekte ve sistem iteratif şekilde çalışmaya devam etmektedir. FL süreci tamamlandıktan sonra ise FedFORGET mekanizması devreye girerek belirli istemcilere ait model katkılarının global modelden çıkarılması sağlanmaktadır. Böylece sistem, merkezi veri paylaşımı gerektirmeden hem öğrenme hem de unutma işlemlerini gerçekleştirebilmektedir. Şekil 4.20.'de bulut ortamında gerçekleştirilen iki aşamalı FL ve FU mimarisinin genel yapısı gösterilmektedir.



Şekil 4.20. Google Cloud ortamında gerçekleştirilen FL ve FU mimarisinin genel yapısı.

Şekil 4.20.'ye göre, sistem ilk aşamada FL süreci gerçekleştirilirken, ikinci aşamada istemci düzeyinde FU işlemi uygulanmaktadır. Şeklin üst bölümünde gösterilen FL aşamasında sistem, *T* tur boyunca iteratif olarak çalışmaktadır. Her istemci kendi yerel veri kümesi üzerinde bağımsız şekilde CBA modeli eğitmekte ve elde edilen yerel modeller merkezi sunucuya göndermektedir. Şekilde yer alan D_i^t ifadesi, *i.* istemcinin *t.* turdaki yerel veri kümesini temsil etmektedir. Benzer şekilde M_i^t ifadesi ise *i.* istemcinin *t.* tur sonunda oluşturduğu yerel CBA modelini göstermektedir. Merkezi sunucu tarafında istemcilerden gelen bu yerel modeller FedCBA algoritması ile birleştirilmekte ve birleşik global model oluşturulmaktadır. Şekilde G^t ile gösterilen yapı, *t.* tur sonunda elde edilen global modeli ifade etmektedir. Oluşturulan global model daha sonra tekrar istemcilere dağıtılmakta ve sistem bir sonraki federe öğrenme turuna devam etmektedir.

Şeklin alt bölümünde ise istemci seviyeli FU süreci gösterilmektedir. Bu aşamada herhangi bir istemci sistemden ayrılmak istediğinde, ilgili istemciye ait geçmiş model katkıları merkezi sunucu tarafından izole edilmektedir. Şekilde örnek olarak *k.* istemci unutulacak hedef istemci olarak gösterilmiştir. Merkezi sunucu, istemciye ait saklanan yerel model katkılarını kullanarak ilgili istemcinin etkisini global modelden çıkarmaktadır. Bu işlem sonucunda G^{-k} ile gösterilen yeni global model elde edilmektedir. G^{-k} modeli, unutulması istenen *k.* istemcisinin katkısını içermeyen güncellenmiş global modeli temsil etmektedir.

Ardından yeni model yalnızca sistemde kalan istemcilere dağıtılmakta ve unutulan istemci sistem dışında bırakılmaktadır.

4.6.1. Deney ortamı

Bu çalışma kapsamında geliştirilen sistem, Google Cloud üzerinde çalışan dağıtık bir istemci-sunucu mimarisi şeklinde tasarlanmıştır. Mimari; istemciler, API katmanı ve merkezi sunucu olmak üzere üç temel bileşenden oluşmaktadır. İstemciler, yerel veri kümeleri üzerinde bağımsız şekilde CBA modeli eğitmekte ve elde edilen yerel modeller API aracılığıyla merkezi sunucuya göndermektedir. Merkezi sunucu ise istemcilerden gelen modelleri FedCBA algoritması ile birleştirerek global modeli oluşturmaktadır. Sistem, çok turlu federe öğrenme mantığına uygun biçimde iteratif olarak çalışmaktadır.

İstemci ve sunucu arasındaki haberleşme API tabanlı olarak gerçekleştirilmiştir. İstemciler doğrudan merkezi sunucu ile iletişim kurmak yerine HTTP istekleri aracılığıyla API katmanına bağlanmaktadır. API, istemcilere eğitim sürecinde kullanılacak destek, güven, istemci kimliği ve tur bilgileri gibi yapılandırma parametrelerini iletmektedir. Yerel eğitim tamamlandıktan sonra oluşturulan CBA modelleri tekrar API'ye gönderilmektedir. API katmanı, farklı istemcilerden gelen modelleri geçici olarak liste içerisinde toplamakta ve belirlenen istemci sayısına ulaşıldığında bu modelleri merkezi sunucuya aktarmaktadır. API ile merkezi sunucu arasındaki veri aktarımında WebSocket tabanlı haberleşme kullanılmıştır. Böylece istemci ve sunucu arasında sürekli ve hızlı veri iletişimi sağlanmıştır.

Merkezi sunucu tarafında, istemcilerden gelen yerel modeller FedCBA yöntemi ile birleştirilmekte ve birleşik global model oluşturulmaktadır. Oluşturulan global model, test veri kümesi üzerinde değerlendirilerek modelin başarısı hesaplanmaktadır. Federe öğrenme süreci tamamlandıktan sonra ise sistem unutma fazına geçmektedir. Bu aşamada belirli bir istemcinin sistemden çıkarılması veya modele olan katkısının kaldırılması talep edildiğinde, ilgili istemcinin katkısı FedFORGET mekanizması ile global modelden çıkarılmaktadır. Böylece unutmuş yeni bir global model elde edilmektedir.

Bu mimaride, istemci katkılarının izlenebilirliğini sağlamak amacıyla merkezi sunucu, istemcilerden gelen modelleri istemci kimlikleri ile ilişkilendirerek saklamaktadır. Geleneksel yaklaşımda her istemcinin her turda oluşturduğu model ayrı ayrı tutulmaktadır. Ancak bu durum, tur sayısı ve istemci sayısı arttıkça yüksek bellek maliyetine neden olmaktadır. Bu çalışmada ise her istemci için tek bir kümülatif model yapısı kullanılmıştır. Böylece istemcinin tüm model katkıları birleşik bir yapı içerisinde temsil edilmekte ve istemci düzeyindeki unutma işlemleri daha verimli gerçekleştirilebilmektedir. Ayrıca bu

yaklaşım, sistemin bellek maliyetini azaltarak mimarinin ölçeklenebilirliğini artırmaktadır. Google Cloud ortamında gerçekleştirilen üç katmanlı istemci–API–sunucu mimarisinin çalışma akışı aşağıdaki adımlardan oluşmaktadır:

1. API sunucusu, istemcilere eğitim sürecinde kullanılacak yapılandırma bilgilerini göndermektedir. Bu bilgiler; veri seti bilgisi, destek ve güven eşik değerleri, toplam tur sayısı ve istemci kimliği gibi parametreleri içermektedir.
2. Her istemci, kendi yerel veri kümesi üzerinde bağımsız şekilde CBA algoritmasını kullanarak yerel modelini eğitmektedir. Bu aşamada veriler istemci cihazını terk etmemektedir.
3. Eğitilen yerel CBA modelleri istemciler tarafından API sunucusuna gönderilmektedir. Her model ilgili istemcinin mevcut turdaki sürüm bilgisini içermektedir.
4. API sunucusu, istemcilerden gelen modelleri geçici olarak listelemekte ve gerekli istemci sayısına ulaşıldığında modeller toplu şekilde merkezi sunucuya aktarmaktadır. Böylece model aktarımı kontrollü ve senkronize biçimde gerçekleştirilmektedir.
5. Merkezi sunucu, istemcilerden gelen yerel CBA modellerini FedCBA algoritması ile birleştirerek global modeli oluşturmaktadır. Bu aşamada istemcilerden gelen ilişkisel kurallar bir araya getirilmekte ve birleşik global kural kümesi elde edilmektedir.
6. Oluşturulan global model, merkezi sunucu üzerinde bulunan değerlendirme katmanında test veri kümesi kullanılarak analiz edilmektedir. Böylece her FL turunun sonunda global modelin performansı izlenebilmektedir.
7. Değerlendirme işlemi tamamlandıktan sonra oluşturulan global model merkezi sunucu tarafından API sunucusuna iletilmektedir.
8. API sunucusu, versiyon kontrolü gerçekleştirerek en güncel global modeli istemcilere dağıtmaktadır. Böylece tüm istemcilerin aynı model sürümü ile çalışması sağlanmaktadır.
9. İstemciler güncellenmiş global modeli aldıktan sonra bir sonraki FL turuna devam etmektedir. Bu süreç belirlenen toplam tur sayısı boyunca iteratif olarak sürdürülmektedir.
10. FL eğitim süreci tamamlandıktan sonra, sistemden ayrılmak veya verisinin unutulmasını talep eden istemcinin kimliği API sunucusu aracılığıyla merkezi sunucuya gönderilmektedir.
11. Merkezi sunucu üzerinde bulunan model ve geçmiş deposu, her istemciye ait önceki tur modellerini saklamaktadır. Bu depo sayesinde istemcilerin geçmiş katkıları

izlenebilmekte ve unutma işlemleri sırasında ilgili istemciye ait modeller geri çağrılabilmektedir.

12. Merkezi sunucu, geçmiş deposunda saklanan istemci modellerini kullanarak ilgili istemcinin modele olan katkısını FedFORGET mekanizması ile global modelden çıkarmaktadır. Böylece yalnızca hedef istemcinin etkisi sistemden kaldırılmaktadır.
13. İstemci katkısının çıkarılması sonucunda yeni bir global model oluşturulmakta ve güncellenmiş model yalnızca sistemde kalan istemcilere dağıtılmaktadır. Böylece istemci düzeyinde federe unutma işlemi tamamlanmaktadır.

4.6.2. Deneysel Bulgular

Deneysel çalışmalar, önerilen FedCBA ve FedFORGET mimarisinin Google Cloud tabanlı dağıtık ortam üzerindeki davranışını analiz edebilmek amacıyla IBM HR veri seti kullanılarak gerçekleştirilmiştir. Bu bölümde elde edilen sonuçlar, önceki simülasyon ve fiziksel istemci ortamlarından farklı olarak API destekli istemci-sunucu mimarisi ve çok turlu bulut tabanlı federe yapı üzerinde elde edilmiştir. Bu nedenle elde edilen sonuçlar, klasik tek aşamalı simülasyon ortamlarından farklılık gösterebilmektedir. Çalışmada sistem çok turlu bir yapı üzerinde çalıştırılmıştır. Veri seti ilk olarak istemci sayısına göre bölünmüş, ardından her istemcideki veri kendi içerisinde tur sayısına göre ayrılmıştır. Her turda istemciler ilgili veri parçası üzerinde yeniden yerel model eğitimi gerçekleştirmiş ve oluşan modeller merkezi sunucuya gönderilmiştir. Tur sonunda oluşturulan global model tekrar istemcilere dağıtılarak bir sonraki eğitim süreci başlatılmıştır. Bu iteratif yapı nedeniyle her turda modelin güncellenmesi, veri parçalarının değişmesi ve yeniden eğitim süreçlerinin gerçekleştirilmesi performans sonuçlarının farklılaşmasına neden olabilmektedir.

Deneysel senaryoların özeti Tablo 4.22.'de sunulmaktadır. Çalışmada IID ve Non-IID olmak üzere iki farklı veri dağılımı senaryosu ele alınmıştır. İlk senaryoda veri seti departman bilgilerine göre istemcilere ayrılmış ve heterojen veri yapısını temsil eden Non-IID dağılım oluşturulmuştur. Non-IID yapıda unutma işlemleri her istemci için ayrı ayrı gerçekleştirilmiş ve her departmanın modele olan katkısı bağımsız olarak analiz edilmiştir. İkinci senaryoda ise veri seti istemciler arasında rastgele ve dengeli biçimde dağıtılarak IID veri yapısı oluşturulmuştur. IID senaryosunda unutma işlemi hedef istemci olarak belirlenen 1 numaralı istemci üzerinde uygulanmıştır. Böylece önerilen mimarinin hem heterojen hem de homojen veri dağılımları altındaki performansı analiz edilmiştir.

Tablo 4.22. Google Cloud ortamında gerçekleştirilen deneysel senaryoların özeti.

Senaryo	Veri Dağılımı	İstemci Sayısı	Unutulacak Hedef İstemci	Açıklama
Senaryo 1	Non-IID	3 (Ar-Ge, İK, Satış)	Ar-Ge İK Satış	Veri seti departmanlara göre bölünmüştür.
Senaryo 2	IID	3 (İstemci 1, İstemci 2, İstemci 3)	İstemci 1	Veri seti istemciler arasında rastgele ve dengeli biçimde bölünmüştür.

Deneysel sonuçlar, modelin öğrenme ve unutma performansını değerlendirebilmek amacıyla farklı ölçümler üzerinden analiz edilmiştir. CBA Test ölçümü, merkezi ortamda eğitilen CBA modelinin test verisi üzerindeki doğruluğunu göstermektedir. FedCBA Test ölçümü, federe öğrenme süreci sonunda oluşturulan global modelin test verisi üzerindeki doğruluk sonucunu ifade etmektedir. FedCBA Hedef ölçümü, unutma işlemi uygulanmadan önce hedef istemci verisi üzerindeki doğruluk değerini göstermektedir. FedFORGET Hedef ölçümü, unutma işlemi sonrasında hedef istemci verisi üzerindeki doğruluk sonucunu ifade etmektedir. FedFORGET Test ölçümü ise unutma işlemi tamamlandıktan sonra global modelin genel test verisi üzerindeki doğruluğunu göstermektedir. Bu ölçümler sayesinde modelin hem öğrenme başarısı hem de unutma işleminin etkinliği birlikte değerlendirilebilmektedir. Tablo 4.23.'de Non-IID veri dağılımı altında her istemci için ayrı ayrı FedCBA ve FedFORGET performans sonuçları yer almaktadır.

Tablo 4.23. İki turlu Non-IID veri dağılımı altında her istemci için ayrı ayrı gerçekleştirilen FedCBA ve FedFORGET performans sonuçları.

		Federe Öğrenme Turu ve Model Birleştirme			**İkinci Federe Tur Sonrası Unutma Süreci**	
Tur	İstemci	CBA Test	FedCBA Hedef	FedCBA Test	FedFORGET Hedef	FedFORGET Test
1	İK	0,58	0, 89	0,67	-	-
	Ar-Ge	0,63	0,71		-	-
	Satış	0,61	0,68		-	-
2	İK	0,54	0, 80	0,69	0,69	0,71
	Ar-Ge	0,66	0,69		0,66	0,67
	Satış	0,69	0,74		0,65	0,54

Tablo 4.23'te Non-IID veri dağılımı altında gerçekleştirilen iki turlu FedCBA ve FedFORGET süreçlerine ait performans sonuçları sunulmaktadır. Bu senaryoda veri seti departman bazlı olarak istemcilere ayrılmış ve İK, Ar-Ge ve Satış departmanları bağımsız istemciler olarak değerlendirilmiştir. Çalışmada unutma işlemleri sıralı veya birbirine bağlı şekilde gerçekleştirilmemiştir. Her istemci için unutma süreci bağımsız olarak uygulanmış

ve ilgili istemcinin modele olan katkısı ayrı ayrı analiz edilmiştir. Böylece her departmanın global model üzerindeki etkisi bağımsız biçimde değerlendirilebilmiştir. Tabloda bulunan "-" işareti, ilgili turda unutma işleminin uygulanmadığını ifade etmektedir.

Birinci tur sonuçları incelendiğinde, FedCBA Test doğruluklarının tüm istemciler için aynı olduğu görülmektedir. Bu durum, farklı veri dağılımlarına sahip istemcilerden elde edilen yerel modellerin FedCBA algoritması ile başarılı şekilde birleştirilebildiğini göstermektedir.

İkinci turda ise her istemci için bağımsız FedFORGET süreçleri uygulanmıştır. İK istemcisi için gerçekleştirilen unutma işleminde FedFORGET Hedef doğruluğunun düşmesi, ilgili istemcinin modele olan katkısının başarıyla çıkarıldığını göstermektedir. Benzer şekilde Ar-Ge ve Satış istemcileri için gerçekleştirilen unutma işlemlerinde de hedef doğruluk değerlerinde azalma gözlemlenmiştir. Bu durum, FedFORGET mekanizmasının farklı istemciler üzerinde etkili biçimde çalışabildiğini göstermektedir.

FedFORGET Test sonuçları incelendiğinde ise unutma işlemi sonrasında global model performansının tamamen bozulmadığı görülmektedir. Özellikle İK ve Ar-Ge istemcilerinde genel test performansının büyük ölçüde korunabildiği gözlemlenmiştir. Satış istemcisinde ise diğer istemcilere kıyasla daha belirgin bir performans düşüşü oluşmuştur. Bunun temel nedeni, Non-IID veri yapısında istemcilerin birbirinden farklı veri dağılımlarına sahip olması ve bazı istemcilerin global modele daha yüksek düzeyde katkı sağlamasıdır.

Genel olarak değerlendirildiğinde, elde edilen sonuçlar önerilen FedFORGET yaklaşımının Non-IID veri dağılımı altında istemci düzeyinde unutma işlemini başarıyla gerçekleştirebildiğini göstermektedir. Ayrıca unutma işlemlerinin her istemci için bağımsız şekilde uygulanabilmesi, yöntemin esnek ve ölçeklenebilir bir yapı sunduğunu ortaya koymaktadır. Tablo 4.24.'de iki turlu IID veri dağılımı altında her istemci için FedCBA ve İstemci 1 için FedFORGET performans sonuçları yer almaktadır

Tablo 4.24. İki turlu IID veri dağılımı altında her istemci için FedCBA ve İstemci 1 için FedFORGET performans sonuçları.

		Federe Öğrenme Turu ve Model Birleştirme			**İkinci Federe Tur Sonrası Unutma Süreci**	
Tur	İstemci	CBA Test	FedCBA Hedef	FedCBA Test	FedFORGET Hedef	FedFORGET Test
1	İstemci 1	0,70	0,74	0,67	-	-
	İstemci 2	0,67	0,73		-	-
	İstemci 3	0,73	0,72		-	-
	İstemci 1	0,71	0,72		0,68	0,68

2	İstemci 2	0,71	0,71	0,69	-	-
	İstemci 3	0,71	0,72		-	-

Tablo 4.24'te IID veri dağılımı altında gerçekleştirilen iki turlu FedCBA ve FedFORGET süreçlerine ait performans sonuçları sunulmaktadır. Bu senaryoda veri seti istemciler arasında rastgele ve dengeli biçimde dağıtılmıştır. Böylece tüm istemcilerin benzer veri özelliklerine sahip olduğu homojen bir yapı oluşturulmuştur. Non-IID senaryosundan farklı olarak unutma işlemi yalnızca İstemci 1 üzerinde uygulanmıştır. İstemci 2 ve İstemci 3 için ise yalnızca FL performans sonuçları değerlendirilmiştir. Tabloda bulunan "-" işareti, ilgili turda unutma işleminin uygulanmadığını ifade etmektedir.

Birinci tur sonuçları incelendiğinde, tüm istemciler için FedCBA Test doğruluklarının aynı olduğu görülmektedir. Bu durum, IID veri dağılımında istemcilerin benzer veri özelliklerine sahip olması nedeniyle global modelin daha dengeli öğrenme gerçekleştirdiğini göstermektedir.

İkinci turda İstemci 1 için FedFORGET süreci uygulanmıştır. Unutma işlemi sonrasında FedFORGET Hedef doğruluğunun FedCBA Hedef değerine göre düşmesi, hedef istemcinin modele olan katkısının başarıyla çıkarıldığını göstermektedir. Bununla birlikte FedFORGET Test sonucunun yüksek seviyede kalması, unutma işlemi sonrasında global modelin genel öğrenme kapasitesini büyük ölçüde koruyabildiğini göstermektedir.

IID veri yapısında istemciler arasında veri dağılımı benzer olduğu için model performansının daha stabil olduğu gözlemlenmiştir. Genel olarak değerlendirildiğinde, önerilen FedFORGET yaklaşımının IID veri dağılımı altında da etkili şekilde çalıştığı ve hedef istemci katkısını modelden başarılı biçimde çıkarabildiği görülmektedir.

5. TARTIŞMA VE SONUÇ

Bu bölümde, tez kapsamında geliştirilen FedFORGET yaklaşımının elde edilen deneysel sonuçları genel hatlarıyla özetlenmektedir. Yöntemin doğruluk, unutma başarımı ve işlem süresi açısından elde ettiği sonuçlara yer verilmekte, ayrıca sistemin güçlü yönleri ve sınırlılıkları anlatılmaktadır. Bunun yanı sıra, çalışmanın kapsamı doğrultusunda gelecekte yapılabilecek geliştirmeler ve araştırma yönleri de bu bölümde ele alınmaktadır.

5.1. Elde Edilen Bulguların Değerlendirilmesi ve Tartışılması

Bu tez kapsamında geliştirilen FedFORGET yaklaşımı, FL ortamında istemci bazlı makine unutmasını etkin bir şekilde gerçekleştirebildiğini göstermiştir. Elde edilen bulgular, önerilen yöntemin hem doğruluk hem de işlem süresi açısından güçlü bir performans sunduğunu ortaya koymaktadır.

Deneysel sonuçlara göre, FedFORGET yöntemi hedef istemciye ait bilgileri global modelden kaldırabilmektedir. Unutma sonrası elde edilen sonuçların, yeniden eğitim yaklaşımı ile aynı olduğu gözlemlenmiştir. FedCBA ile elde edilen global model performansının korunması, önerilen yöntemin model bütünlüğünü bozmadığını ortaya

koymaktadır. Özellikle Non-IID veri dağılımı altında elde edilen sonuçlar, yöntemin gerçek dünya senaryolarında da uygulanabilir olduğunu göstermektedir. Ayrıca, istemci sayısının artırıldığı deneylerde yöntemin ölçeklenebilir bir yapı sergilediği gözlemlenmiştir. İstemci sayısı arttıkça performansta ciddi bir düşüş yaşanmamıştır. Bu durum, FedFORGET yaklaşımının büyük ölçekli sistemlerde de kullanılabileceğini göstermektedir.

Çalışma yalnızca simülasyon ortamı ile sınırlı kalmamış, aynı zamanda Google Cloud ve Raspberry Pi ortamlarında da uygulanmıştır. Bu ortamlarda elde edilen sonuçlar, önerilen yöntemin hedef istemciye ait bilgileri doğru ve tutarlı bir şekilde modelden çıkarabildiğini göstermektedir. Böylece, FedFORGET yaklaşımının farklı ortamlarda uygulanabilir olduğu ortaya konmuştur.

5.2. Sistemin Avantajları ve Kısıtları

Önerilen FedFORGET yaklaşımının temel avantajlarından biri, unutma işlemini veriye erişim gerektirmeden doğrudan model üzerinde algoritmik olarak gerçekleştirmesidir. Bu durum, özellikle veri gizliliğinin kritik olduğu senaryolarda önemli bir avantaj sağlamaktadır. Bunun yanı sıra, yöntemin CBA tabanlı ilişkisel sınıflandırma yapısına dayanması, modelin doğası gereği yorumlanabilir olmasını sağlamaktadır. Kural tabanlı yapı sayesinde, modelden çıkarılan bilginin hangi kurallar üzerinden etki ettiği açık bir şekilde izlenebilmekte ve doğrulanabilmektedir. Bu özellik, kara kutu yaklaşımlara kıyasla önemli bir üstünlük sunmaktadır.

Yöntemin başlıca kısıtlarından biri, yalnızca istemci bazlı unutma senaryosu için tasarlanmış olmasıdır. Örnek veya sınıf etiketi bazlı unutma, bu çalışma kapsamında ele alınmamıştır. Diğer bir kısıt, yöntemin modele özgü bir yapıda olması ve ilişkisel sınıflandırma algoritması CBA'ya dayanmasıdır. Bu durum, yaklaşımın farklı model türlerine genellenebilirliğini sınırlamaktadır.

5.3. Gelecek Çalışmalar için Öneriler

Gelecek çalışmalarda, önerilen FedFORGET yaklaşımının farklı ilişkisel sınıflandırma algoritmaları ile genişletilmesi planlanmaktadır. Özellikle CBA dışında FCBA ve WCBA gibi yöntemlerin kullanılması, model performansını artırabilir ve farklı veri yapılarında daha güçlü sonuçlar elde edilmesini sağlayabilir.

Bu çalışmada yalnızca istemci bazlı unutma ele alınmıştır. Gelecek çalışmalarda, örnek bazlı ve sınıf etiketi bazlı unutma yaklaşımlarına odaklanılması hedeflenmektedir.

Ayrıca, yöntemin farklı veri setleri ve daha büyük ölçekli federe sistemler üzerinde test edilmesi, genellenebilirliğinin daha kapsamlı şekilde değerlendirilmesini sağlayacaktır.

Son olarak, önerilen yaklaşımın gerçek zamanlı sistemlere entegrasyonu ve farklı uygulama alanlarında kullanımı gelecekte araştırılması gereken önemli konular arasında yer almaktadır.

KAYNAKLAR/REFERENCES

1. Buyuktanir, B., et al., *Enhancing Employee Retention Prediction Through Federated Learning and Associative Classification using FedCBA: A Case Study on IBM HR Analytics.* Arabian Journal for Science and Engineering, 2026: p. 1–28.
2. Telikani, A., et al., *Evolutionary machine learning: A survey.* ACM Computing Surveys (CSUR), 2021. **54**(8): p. 1–35.
3. Atalay, M. and E. Çelik, *Büyük veri analizinde yapay zekâ ve makine öğrenmesi uygulamalari-artificial intelligence and machine learning applications in big data analysis.* Mehmet Akif Ersoy University Journal of Social Sciences Institute, 2017. **9**(22): p. 155–172.
4. Nergiz, M., *İşbirlikçi yapay zeka konsepti: federe öğrenmeye genel bir bakış.* Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 2022. **13**(2): p. 279–286.
5. EFENDİOĞLU, İ.H., *Tüketici Verilerinin Ticarileştirilmesi ve Dijital Etik.* Yapay Zekâ Uygulamaları: Sektörel Etkiler ve: p. 11.
6. Başkaya, F. and H. Karacan, *Yapay zekâ tabanlı sistemlerin kişisel veri mahremiyeti üzerine etkisi: Sohbet robotları üzerine inceleme.* Bilişim Teknolojileri Dergisi, 2022. **15**(4): p. 481–491.
7. Balaban, M.F. and M.A.K. Ibrahım, *SOSYAL MEDYA VE UNUTULMA HAKKI: META THREADS ÖRNEĞİ.* Bilişim Hukuku Dergisi, 2023. **5**(2): p. 281–338.
8. Buyuktanir, B., et al., *Federated learning in intrusion detection: advancements, applications, and future directions.* Cluster Computing, 2025. **28**(7).
9. Wu, H. and P. Wang, *Fast-convergent federated learning with adaptive weighting.* IEEE Transactions on Cognitive Communications and Networking, 2021. **7**(4): p. 1078–1088.
10. Campos, E.M., et al., *Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges.* Computer Networks, 2022. **203**: p. 108661.
11. Dayal, S., et al. *Comparative analysis of membership inference attacks in federated learning*. in *Proceedings of the 27th International Database Engineered Applications Symposium*. 2023.
12. Hatamizadeh, A., et al., *Do gradient inversion attacks make federated learning unsafe?* IEEE Transactions on Medical Imaging, 2023. **42**(7): p. 2044–2056.
13. Liu, Y., et al. *The right to be forgotten in federated learning: An efficient realization with rapid retraining*. in *IEEE INFOCOM 2022-IEEE conference on computer communications*. 2022. IEEE.
14. Zhao, Y., et al., *Federated learning with non-iid data.* arXiv preprint arXiv:1806.00582, 2018.
15. Buyuktanir, B., K. Yildiz, and G.K. Baydogmus, *A Systematic Mapping Study on Machine Unlearning in Federated Learning*, in *2025 7th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (ICHORA)*. 2025. p. 1–10.
16. Liu, Z., et al., *A survey on federated unlearning: Challenges, methods, and future directions.* ACM Computing Surveys, 2024. **57**(1): p. 1–38.
17. Zhao, Y., et al., *Exploring federated unlearning: Review, comparison, and insights.* IEEE Network, 2025.
18. Adadi, A. and M. Berrada, *Peeking inside the black-box: a survey on explainable artificial intelligence (XAI).* IEEE access, 2018. **6**: p. 52138–52160.
19. Chinnaraju, A., *Explainable AI (XAI) for trustworthy and transparent decision-*

making: A theoretical framework for AI interpretability. World Journal of Advanced Engineering Technology and Sciences, 2025. **14**(3): p. 170–207.
20. Tosun, B.E., et al. *Comparative analysis of relational classification algorithms on benchmark datasets*. in *2025 10th International Conference on Computer Science and Engineering (UBMK)*. 2025. IEEE.
21. Büyüktanir, B., et al., *du-CBA: Data-agnostic and incremental classification-based association rules extraction architecture du-CBA: Veriden habersiz ve artirimli siniflandirmaya dayali birliktelik kurallari çikarma mimarisi.* Journal of the Faculty of Engineering and Architecture of Gazi University, 2023. **38**(3).
22. Mantelero, A., *The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'.* Computer Law & Security Review, 2013. **29**(3): p. 229–235.
23. Gu, J., et al., *Recent advances in convolutional neural networks.* Pattern recognition, 2018. **77**: p. 354–377.
24. Cao, Y. and J. Yang. *Towards making systems forget with machine unlearning*. in *2015 IEEE symposium on security and privacy*. 2015. IEEE.
25. Bourtoule, L., et al. *Machine unlearning*. in *2021 IEEE symposium on security and privacy (SP)*. 2021. IEEE.
26. Yan, H., et al. *ARCANE: An Efficient Architecture for Exact Machine Unlearning*. in *Ijcai*. 2022.
27. Golatkar, A., et al. *Mixed-privacy forgetting in deep networks*. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021.
28. Graves, L., V. Nagisetty, and V. Ganesh. *Amnesiac machine learning*. in *Proceedings of the AAAI conference on artificial intelligence*. 2021.
29. Liu, B., Q. Liu, and P. Stone. *Continual learning and private unlearning*. in *Conference on Lifelong Learning Agents*. 2022. PMLR.
30. Koh, P.W. and P. Liang. *Understanding black-box predictions via influence functions*. in *International conference on machine learning*. 2017. PMLR.
31. Ginart, A., et al., *Making ai forget you: Data deletion in machine learning.* Advances in neural information processing systems, 2019. **32**.
32. Nguyen, T.T., et al., *A survey of machine unlearning.* ACM Transactions on Intelligent Systems and Technology, 2025. **16**(5): p. 1–46.
33. Kurt, A., et al., *EVALUATING SISA-BASED MACHINE UNLEARNING ACROSS DIVERSE MODALITIES: TABULAR, VISUAL AND AUDITORY DATA.* Journal of Naval Sciences and Engineering, (Advanced Online Publication): p. 151–187.
34. Li, N., et al., *Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects.* IEEE Transactions on Neural Networks and Learning Systems, 2025.
35. Zhang, B., et al., *Towards certified unlearning for deep neural networks.* arXiv preprint arXiv:2408.00920, 2024.
36. Gupta, V., et al., *Adaptive machine unlearning.* Advances in Neural Information Processing Systems, 2021. **34**: p. 16319–16330.
37. Fraboni, Y., et al. *Sifu: Sequential informed federated unlearning for efficient and provable client unlearning in federated optimization*. in *International Conference on Artificial Intelligence and Statistics*. 2024. PMLR.
38. Golatkar, A., A. Achille, and S. Soatto. *Eternal sunshine of the spotless net: Selective forgetting in deep networks*. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.
39. Neel, S., A. Roth, and S. Sharifi-Malvajerdi. *Descent-to-delete: Gradient-based methods for machine unlearning*. in *Algorithmic Learning Theory*. 2021. PMLR.
40. Ullah, E., et al. *Machine unlearning via algorithmic stability*. in *Conference on*

Learning Theory. 2021. PMLR.
41. Chen, Y., et al., *A novel online incremental and decremental learning algorithm based on variable support vector machine.* Cluster Computing, 2019. **22**(Suppl 3): p. 7435–7445.
42. Chundawat, V.S., et al. *Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher*. in *Proceedings of the AAAI Conference on Artificial Intelligence*. 2023.
43. Kim, H., S. Lee, and S.S. Woo. *Layer attack unlearning: Fast and accurate machine unlearning via layer level attack and knowledge distillation*. in *Proceedings of the AAAI Conference on Artificial Intelligence*. 2024.
44. Wang, L., et al. *Kga: A general machine unlearning framework based on knowledge gap alignment*. in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2023.
45. Zhang, X., et al. *Machine unlearning methodology based on stochastic teacher network.* in *International Conference on Advanced Data Mining and Applications*. 2023. Springer.
46. Nguyen, Q.P., et al. *Markov chain monte carlo-based machine unlearning: Unlearning what needs to be forgotten*. in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*. 2022.
47. Baumhauer, T., P. Schöttle, and M. Zeppelzauer, *Machine unlearning: Linear filtration for logit-based classifiers.* Machine Learning, 2022. **111**(9): p. 3203–3226.
48. Izzo, Z., et al. *Approximate data deletion from machine learning models*. in *International conference on artificial intelligence and statistics*. 2021. PMLR.
49. Li, Y., C.-H. Wang, and G. Cheng, *Online forgetting process for linear regression models.* arXiv preprint arXiv:2012.01668, 2020.
50. Schelter, S., S. Grafberger, and T. Dunning. *Hedgecut: Maintaining randomised trees for low-latency machine unlearning*. in *Proceedings of the 2021 International Conference on Management of Data*. 2021.
51. Wu, Z., et al., *Deltaboost: Gradient boosting decision trees with efficient machine unlearning.* Proceedings of the ACM on Management of Data, 2023. **1**(2): p. 1–26.
52. Basu, S., P. Pope, and S. Feizi, *Influence functions in deep learning are fragile.* arXiv preprint arXiv:2006.14651, 2020.
53. Golatkar, A., A. Achille, and S. Soatto. *Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations*. in *European Conference on Computer Vision*. 2020. Springer.
54. Goyal, A., V. Hassija, and V.H.C.d. Albuquerque. *Revisiting machine learning training process for enhanced data privacy*. in *Proceedings of the 2021 Thirteenth International Conference on Contemporary Computing*. 2021.
55. He, Y., et al., *Deepobliviate: a powerful charm for erasing data residual memory in deep neural networks.* arXiv preprint arXiv:2105.06209, 2021.
56. Mehta, R., et al. *Deep unlearning via randomized conditionally independent hessians*. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.
57. Zhang, P.-F., et al. *Machine unlearning for image retrieval: A generative scrubbing approach*. in *Proceedings of the 30th ACM international conference on multimedia*. 2022.
58. Aldaghri, N., H. Mahdavifar, and A. Beirami, *Coded machine unlearning.* IEEE Access, 2021. **9**: p. 88137–88150.
59. Dukler, Y., et al. *Safe: Machine unlearning with shard graphs*. in *Proceedings of the IEEE/CVF international conference on computer vision*. 2023.

60. Huang, H., et al., *Unlearnable examples: Making personal data unexploitable.* arXiv preprint arXiv:2101.04898, 2021.
61. Shan, S., et al. *Fawkes: Protecting privacy against unauthorized deep learning models*. in *29th USENIX security symposium (USENIX Security 20)*. 2020.
62. Tarun, A.K., et al., *Fast yet effective machine unlearning.* IEEE transactions on neural networks and learning systems, 2023. **35**(9): p. 13046–13055.
63. Yu, D., et al. *How does data augmentation affect privacy in machine learning?* in *Proceedings of the AAAI conference on artificial intelligence*. 2021.
64. Cao, Z., et al. *Machine unlearning method based on projection residual*. in *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*. 2022. IEEE.
65. Peste, A., D. Alistarh, and C.H. Lampert, *Ssse: Efficiently erasing samples from trained machine learning models.* arXiv preprint arXiv:2107.03860, 2021.
66. Zeng, Y., et al. *Modelpred: A framework for predicting trained model from training data*. in *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. 2023. IEEE.
67. Sommer, D.M., et al., *Towards probabilistic verification of machine unlearning.* arXiv preprint arXiv:2003.04247, 2020.
68. Sommer, D.M., et al., *Athena: Probabilistic verification of machine unlearning.* Proceedings on Privacy Enhancing Technologies, 2022.
69. Khan, M.E. and S. Swaroop, *Knowledge-adaptation priors.* Advances in neural information processing systems, 2021. **34**: p. 19757–19770.
70. Chen, M., et al. *When machine unlearning jeopardizes privacy*. in *Proceedings of the 2021 ACM SIGSAC conference on computer and communications security*. 2021.
71. Thudi, A., et al. *Unrolling sgd: Understanding factors influencing machine unlearning*. in *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. 2022. IEEE.
72. Wu, Y., E. Dobriban, and S. Davidson. *Deltagrad: Rapid retraining of machine learning models*. in *International Conference on Machine Learning*. 2020. PMLR.
73. Schelter, S., *Amnesia-a selection of machine learning models that can forget user data very fast.* suicide, 2020. **8364**(44035): p. 46992.
74. Bonato, J., M. Cotogni, and L. Sabetta. *Is retain set all you need in machine unlearning? restoring performance of unlearned models with out-of-distribution images*. in *European Conference on Computer Vision*. 2024. Springer.
75. Guo, C., et al., *Certified data removal from machine learning models.* arXiv preprint arXiv:1911.03030, 2019.
76. Marchant, N.G., B.I. Rubinstein, and S. Alfeld. *Hard to forget: Poisoning attacks on certified machine unlearning*. in *Proceedings of the AAAI Conference on Artificial Intelligence*. 2022.
77. Brophy, J. and D. Lowd. *Machine unlearning for random forests*. in *International conference on machine learning*. 2021. PMLR.
78. Fu, S., F. He, and D. Tao, *Knowledge removal in sampling-based bayesian inference.* arXiv preprint arXiv:2203.12964, 2022.
79. Fu, S., et al., *Bayesian inference forgetting.* arXiv preprint arXiv:2101.06417, 2021.
80. Dai, D., et al. *Knowledge neurons in pretrained transformers*. in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2022.
81. Lin, S., et al. *Erm-ktp: Knowledge-level machine unlearning via knowledge transfer*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*

Recognition. 2023.
82. Cong, W. and M. Mahdavi. *Efficiently forgetting what you have learned in graph representation learning via projection*. in *International Conference on Artificial Intelligence and Statistics*. 2023. PMLR.
83. Cong, W. and M. Mahdavi, *Grapheditor: An efficient graph representation learning and unlearning approach.* 2022.
84. Zhu, X., G. Li, and W. Hu. *Heterogeneous federated knowledge graph embedding learning and unlearning*. in *Proceedings of the ACM web conference 2023*. 2023.
85. Wang, C.-L., M. Huai, and D. Wang. *Inductive graph unlearning*. in *32nd USENIX Security Symposium (USENIX Security 23)*. 2023.
86. Cheng, J., et al., *Gnndelete: A general strategy for unlearning in graph neural networks.* arXiv preprint arXiv:2302.13406, 2023.
87. Chen, M., et al. *Graph unlearning*. in *Proceedings of the 2022 ACM SIGSAC conference on computer and communications security*. 2022.
88. Pan, C., E. Chien, and O. Milenkovic. *Unlearning graph classifiers with limited data resources*. in *Proceedings of the ACM Web Conference 2023*. 2023.
89. Chen, C., et al. *Recommendation unlearning*. in *Proceedings of the ACM web conference 2022*. 2022.
90. Ganhör, C., et al. *Unlearning protected user attributes in recommendations with adversarial training*. in *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 2022.
91. Liu, G., et al. *Federaser: Enabling efficient client-level data removal from federated learning models*. in *2021 IEEE/ACM 29th international symposium on quality of service (IWQOS)*. 2021. IEEE.
92. Huynh, T.T., et al. *Fast-fedul: A training-free federated unlearning with provable skew resilience*. in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 2024. Springer.
93. Tao, Y., et al., *Communication efficient and provable federated unlearning.* arXiv preprint arXiv:2401.11018, 2024.
94. Chen, J. and D. Yang. *Unlearn what you want to forget: Efficient unlearning for llms*. in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 2023.
95. Wang, H., et al., *Towards efficient and effective unlearning of large language models for recommendation.* Frontiers of Computer Science, 2025. **19**(3): p. 193327.
96. Hu, X., et al. *Separate the wheat from the chaff: Model deficiency unlearning via parameter-efficient module operation*. in *Proceedings of the AAAI Conference on Artificial Intelligence*. 2024.
97. Pochinkov, N. and N. Schoots, *Dissecting language models: Machine unlearning via selective pruning.* arXiv preprint arXiv:2403.01267, 2024.
98. Lu, X., et al., *Quark: Controllable text generation with reinforced unlearning.* Advances in neural information processing systems, 2022. **35**: p. 27591–27609.
99. Li, G., et al., *Machine unlearning for image-to-image generative models.* arXiv preprint arXiv:2402.00351, 2024.
100. Zhang, Y., et al., *Defensive unlearning with adversarial training for robust concept erasure in diffusion models.* Advances in neural information processing systems, 2024. **37**: p. 36748–36776.
101. Panda, S. and A. Prathosh, *FAST: Feature Aware Similarity Thresholding for weak unlearning in black-box generative models.* IEEE Transactions on Artificial Intelligence, 2024. **6**(4): p. 885–895.
102. Sayyed, A., et al. *CLUE: Bringing Machine Unlearning to Mobile Devices*. in

Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2026.
103. Artelt, A., et al. *Unsupervised unlearning of concept drift with autoencoders*. in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2023. IEEE.
104. Yuan, Y., et al., *Toward efficient and robust federated unlearning in IoT networks.* IEEE Internet of Things Journal, 2024. **11**(12): p. 22081–22090.
105. Faraji, A. and M. Papagelis, *TraceHiding: Scalable Machine Unlearning for Mobility Data.* arXiv preprint arXiv:2509.17241, 2025.
106. Zeng, Y., et al. *Qoseraser: A data erasable framework for web service qos prediction*. in *2023 IEEE International Conference on Software Services Engineering (SSE)*. 2023. IEEE.
107. Chen, K., et al., *Privacy preserving machine unlearning for smart cities.* Annals of Telecommunications, 2024. **79**(1): p. 61–72.
108. Shoji, M., T. Obo, and N. Kubota. *Add-if-silent rule-based growing neural gas for high-density topological structure of unknown objects*. in *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. 2023. IEEE.
109. Liu, S., et al., *Rethinking machine unlearning for large language models.* Nature Machine Intelligence, 2025. **7**(2): p. 181–194.
110. Shen, S., et al., *MT-KD: Multi-Teacher Knowledge Distillation for Privacy-Aware Unlearning in IIoT Intrusion Detection Systems.* IEEE Transactions on Cognitive Communications and Networking, 2026.
111. Fan, J., et al., *Fast model update for IoT traffic anomaly detection with machine unlearning.* IEEE Internet of Things Journal, 2022. **10**(10): p. 8590–8602.
112. Lu, Z., et al. *Fp 2-mia: A membership inference attack free of posterior probability in machine unlearning*. in *International Conference on Provable Security*. 2022. Springer.
113. Hu, H., et al. *Learn what you want to unlearn: Unlearning inversion attacks against machine unlearning*. in *2024 IEEE Symposium on Security and Privacy (SP)*. 2024. IEEE.
114. Di, J.Z., et al. *Hidden poison: Machine unlearning enables camouflaged poisoning attacks*. in *NeurIPS ML Safety Workshop*. 2022.
115. Hu, H., et al., *A duty to forget, a right to be assured? exposing vulnerabilities in machine unlearning services.* arXiv preprint arXiv:2309.08230, 2023.
116. Zhang, Z., et al., *Safe unlearning: A surprisingly effective and generalizable solution to defend against jailbreak attacks.* arXiv preprint arXiv:2407.02855, 2024. **1**(2): p. 3.
117. Ozturk, O., et al., *Differential Privacy in Federated Learning: Mitigating Inference Attacks with Randomized Response.* arXiv preprint arXiv:2509.13987, 2025.
118. Brodzinski, C.E., *Survey of security and data attacks on machine unlearning in financial and e-commerce.* arXiv preprint arXiv:2410.00055, 2024.
119. Goldreich, O., *Secure multi-party computation.* Manuscript. Preliminary version, 1998. **78**(110): p. 1–108.
120. Weng, J., et al., *Proof of unlearning: Definitions and instantiation.* IEEE Transactions on Information Forensics and Security, 2024. **19**: p. 3309–3323.
121. Goel, S., et al., *Corrective machine unlearning.* arXiv preprint arXiv:2402.14015, 2024.
122. Martínez Llamas, J., D. Preuveneers, and W. Joosen, *Certified unlearning for a trustworthy machine learning-based access control administration.* International Journal of Information Security, 2025. **24**(2): p. 94.

123. Lee, J., B. Bagheri, and H.-A. Kao, *A cyber-physical systems architecture for industry 4.0-based manufacturing systems.* Manufacturing letters, 2015. **3**: p. 18–23.
124. Zhou, K., T. Liu, and L. Zhou. *Industry 4.0: Towards future industrial opportunities and challenges*. in *2015 12th International conference on fuzzy systems and knowledge discovery (FSKD)*. 2015. IEEE.
125. Gökçay, B. and B. Arda, *Kişisel sağlık verilerinin korunması kapsamında sağlık araştırmalarında etik bakış.* Archives of the Turkish Society of Cardiology/Türk Kardiyoloji Derneği Arşivi, 2019. **47**(3).
126. Jiang, J.C., et al., *Federated learning in smart city sensing: Challenges and opportunities.* Sensors, 2020. **20**(21): p. 6230.
127. Kairouz, P. and H.B. McMahan, *Advances and open problems in federated learning.* Foundations and trends in machine learning, 2021. **14**(1-2): p. 1–210.
128. Zhang, C., et al., *A survey on federated learning.* Knowledge-Based Systems, 2021. **216**: p. 106775.
129. Süzen, A.A. and K. Kayaalp, *Büyük verilerde gizlilik tabanli yaklaşim: Federe öğrenme.* International Journal of 3d Printing Technologies and Digital Industry, 2019. **3**(3): p. 297–304.
130. Huang, C., J. Huang, and X. Liu, *Cross-silo federated learning: Challenges and opportunities.* arXiv preprint arXiv:2206.12949, 2022.
131. Huang, Y., et al. *Personalized cross-silo federated learning on non-iid data*. in *Proceedings of the AAAI conference on artificial intelligence*. 2021.
132. ur Rehman, M.H., et al., *TrustFed: A framework for fair and trustworthy cross-device federated learning in IIoT.* IEEE Transactions on Industrial Informatics, 2021. **17**(12): p. 8485–8494.
133. Hahn, C., et al., *Versa: Verifiable secure aggregation for cross-device federated learning.* IEEE Transactions on Dependable and Secure Computing, 2021. **20**(1): p. 36–52.
134. AbdulRahman, S., et al., *A survey on federated learning: The journey from centralized to distributed on-site learning and beyond.* IEEE internet of things journal, 2020. **8**(7): p. 5476–5497.
135. Zhang, H., K. Zeng, and S. Lin, *FedUR: Federated learning optimization through adaptive centralized learning optimizers.* IEEE Transactions on Signal Processing, 2023. **71**: p. 2622–2637.
136. Beltrán, E.T.M., et al., *Fedstellar: A platform for decentralized federated learning.* Expert Systems with Applications, 2024. **242**: p. 122861.
137. Beltrán, E.T.M., et al., *Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges.* IEEE Communications Surveys & Tutorials, 2023. **25**(4): p. 2983–3013.
138. Bonawitz, K., et al., *Towards federated learning at scale: System design.* Proceedings of machine learning and systems, 2019. **1**: p. 374–388.
139. McMahan, B., et al. *Communication-efficient learning of deep networks from decentralized data*. in *Artificial intelligence and statistics*. 2017. Pmlr.
140. Sprague, M.R., et al. *Asynchronous federated learning for geospatial applications*. in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 2018. Springer.
141. 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. **35**(4): p. 3347–3366.
142. Gepperth, A. and B. Hammer. *Incremental learning algorithms and applications*. in *European symposium on artificial neural networks (ESANN)*. 2016.

143. Önlü, A.Ö., et al. *An investigation and performance evaluation of aggregation algorithms in federated learning architecture*. in *2025 Innovations in Intelligent Systems and Applications Conference (ASYU)*. 2025. IEEE.
144. Li, X., et al., *On the convergence of fedavg on non-iid data.* arXiv preprint arXiv:1907.02189, 2019.
145. An, T., et al., *Consideration of FedProx in privacy protection.* Electronics, 2023. **12**(20): p. 4364.
146. Wang, H., et al., *Federated learning with matched averaging.* arXiv preprint arXiv:2002.06440, 2020.
147. Wang, J., et al., *Tackling the objective inconsistency problem in heterogeneous federated optimization.* Advances in neural information processing systems, 2020. **33**: p. 7611–7623.
148. Chen, H.-Y. and W.-L. Chao, *Fedbe: Making bayesian model ensemble applicable to federated learning.* arXiv preprint arXiv:2009.01974, 2020.
149. Arivazhagan, M.G., et al., *Federated learning with personalization layers.* arXiv preprint arXiv:1912.00818, 2019.
150. Lu, Z., et al., *Federated learning with non-iid data: A survey.* IEEE Internet of Things Journal, 2024. **11**(11): p. 19188–19209.
151. Zhang, X., et al., *Federated feature selection for horizontal federated learning in IoT networks.* IEEE Internet of Things Journal, 2023. **10**(11): p. 10095–10112.
152. Liu, Y., et al., *Vertical federated learning: Concepts, advances, and challenges.* IEEE transactions on knowledge and data engineering, 2024. **36**(7): p. 3615–3634.
153. Saha, S. and T. Ahmad, *Federated transfer learning: Concept and applications.* Intelligenza Artificiale, 2021. **15**(1): p. 35–44.
154. Wen, J., et al., *A survey on federated learning: challenges and applications.* International journal of machine learning and cybernetics, 2023. **14**(2): p. 513–535.
155. Ouyang, L., et al., *A novel framework of collaborative early warning for COVID-19 based on blockchain and smart contracts.* Information sciences, 2021. **570**: p. 124–143.
156. Dayan, I., et al., *Federated learning for predicting clinical outcomes in patients with COVID-19.* Nature medicine, 2021. **27**(10): p. 1735–1743.
157. Ma, Z., et al., *An assisted diagnosis model for cancer patients based on federated learning.* Frontiers in Oncology, 2022. **12**: p. 860532.
158. Du, Y., et al., *Federated matrix factorization for privacy-preserving recommender systems.* Applied soft computing, 2021. **111**: p. 107700.
159. Caballero, A., et al., *Activity recommendation in intelligent campus environments based on the Eduroam federation.* Journal of ambient intelligence and smart environments, 2016. **8**(1): p. 35–46.
160. Muhammad, K., et al. *Fedfast: Going beyond average for faster training of federated recommender systems*. in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2020.
161. Putra, K.T., et al., *Federated compressed learning edge computing framework with ensuring data privacy for PM2. 5 prediction in smart city sensing applications.* Sensors, 2021. **21**(13): p. 4586.
162. Yuan, X., et al., *Fedstn: Graph representation driven federated learning for edge computing enabled urban traffic flow prediction.* IEEE Transactions on Intelligent Transportation Systems, 2022. **24**(8): p. 8738–8748.
163. Cheng, K., et al., *Secureboost: A lossless federated learning framework.* IEEE intelligent systems, 2021. **36**(6): p. 87–98.
164. Ye, Y., et al., *EdgeFed: Optimized federated learning based on edge computing.*

IEEe Access, 2020. **8**: p. 209191–209198.
165. Jiang, H., et al., *Customized federated learning for accelerated edge computing with heterogeneous task targets.* Computer Networks, 2020. **183**: p. 107569.
166. Wang, Q., et al., *Efficient federated learning for fault diagnosis in industrial cloud-edge computing.* Computing, 2021. **103**(10): p. 2319–2337.
167. Tang, Z., H. Hu, and C. Xu, *A federated learning method for network intrusion detection.* Concurrency and Computation: Practice and Experience, 2022. **34**(10): p. e6812.
168. Zhao, R., et al., *Intelligent intrusion detection based on federated learning aided long short-term memory.* Physical Communication, 2020. **42**: p. 101157.
169. Liu, H., et al., *Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing.* IEEE Transactions on Vehicular Technology, 2021. **70**(6): p. 6073–6084.
170. Jiang, Y., et al., *Towards efficient and certified recovery from poisoning attacks in federated learning.* IEEE Transactions on Information Forensics and Security, 2025.
171. Gao, X., et al., *Verifi: Towards verifiable federated unlearning.* IEEE Transactions on Dependable and Secure Computing, 2024. **21**(6): p. 5720–5736.
172. Dhasade, A., et al., *Quickdrop: Efficient federated unlearning by integrated dataset distillation.* arXiv preprint arXiv:2311.15603, 2023.
173. Wang, W., et al. *Bfu: Bayesian federated unlearning with parameter self-sharing*. in *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*. 2023.
174. El Naqa, I. and M.J. Murphy, *What is machine learning?*, in *Machine learning in radiation oncology: theory and applications*. 2015, Springer. p. 3–11.
175. Liu, B., W. Hsu, and Y. Ma. *Integrating classification and association rule mining.* in *Proceedings of the fourth international conference on knowledge discovery and data mining*. 1998.
176. Kotsiantis, S.B., I.D. Zaharakis, and P.E. Pintelas, *Machine learning: a review of classification and combining techniques.* Artificial Intelligence Review, 2006. **26**(3): p. 159–190.
177. Lorena, A.C., A.C. De Carvalho, and J.M. Gama, *A review on the combination of binary classifiers in multiclass problems.* Artificial Intelligence Review, 2008. **30**(1): p. 19.
178. Hassan, C.A.U., M.S. Khan, and M.A. Shah. *Comparison of machine learning algorithms in data classification*. in *2018 24th international conference on automation and computing (ICAC)*. 2018. IEEE.
179. Agrawal, R., T. Imieliński, and A. Swami. *Mining association rules between sets of items in large databases*. in *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*. 1993.
180. Rana, S., M.N.I. Mondal, and M.R. Islam. *Unveiling purchase patterns: A comparative analysis of Apriori, FP-Growth, and Eclat algorithms in market basket analysis*. in *2025 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. 2025. IEEE.
181. Lobo, D. *Association rules: Normalizing the lift*. in *Ninth International Conference on Digital Information Management (ICDIM 2014)*. 2014. IEEE.
182. Girotra, M., et al., *Comparative survey on association rule mining algorithms.* International Journal of Computer Applications, 2013. **84**(10): p. 14612–2862.
183. Dunham, M.H., et al., *A survey of association rules.* ACM Survey Journal, 2000: p. 1–65.
184. Zhou, D.-W., et al., *Class-incremental learning: A survey.* IEEE Transactions on

Pattern Analysis and Machine Intelligence, 2024. **46**(12): p. 9851–9873.
185. 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.
186. 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.
187. 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.
188. 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.
189. Alnababteh, M.H., et al., *Associative classification based on incremental mining (acim).* International Journal of Computer Theory and Engineering, 2014. **6**(2): p. 135.
190. Al-Fayoumi, M.A., *Enhanced Associative classification based on incremental mining Algorithm (E-ACIM).* International Journal of Computer Science Issues (IJCSI), 2015. **12**(1): p. 124.
191. Jayaraj, A., et al. *Classification of Health of Foetus using Machine Learning*. in *2025 International Conference on Intelligent and Cloud Computing (ICoICC)*. 2025. IEEE.
192. Tan, C.Y., et al., *Amogel: a multi-omics classification framework using associative graph neural networks with prior knowledge for biomarker identification.* BMC bioinformatics, 2025. **26**(1): p. 94.
193. Costa, G. and R. Ortale. *MedRule: An Interpretable Rule Induction Framework for Reliable Clinical Decisions*. in *Proceedings of the 2025 9th International Conference on Computer Science and Artificial Intelligence*. 2025.
194. Demir, A., et al. *Model Training and Real World Analysis Using Health Data with Federated Learning*. in *The International Open Source Conference (UAKK 2024)*. 2025.
195. Boppiniti, S.T., *A survey on explainable AI: Techniques and challenges.* Novateur Publications, International Journal of Innovations in Engineering Research and Technology., 2020.
196. Bhati, D., et al., *A survey of post-hoc xai methods from a visualization perspective: Challenges and opportunities.* IEEE Access, 2025.
197. Yağin, F.H., et al., *Comparison of Performances of Associative Classification Methods for Cervical Cancer Prediction: Observational Study.* Turkiye Klinikleri Journal of Biostatistics, 2021. **13**(3).
198. Maloku, F. and B. Maloku, *Analyzing IBM HR data: Employee attrition and performance insights.* J. Eng. Appl. Sci. Technol. SRC/JEAST-382, 2024. **268**: p. 2–10.
199. Chakrabarty, N. and S. Biswas. *A statistical approach to adult census income level prediction*. in *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. 2018. IEEE.
200. Wibowo, A., et al. *Classification algorithm for edible mushroom identification*. in *2018 International Conference on Information and Communications Technology (ICOIACT)*. 2018. IEEE.
201. Awwalu, J., A. Ghazvini, and A.A. Bakar, *Performance comparison of data mining algorithms: a case study on car evaluation dataset.* Int. J. Comput. Trends Technol,

2014. **13**(2): p. 78–82.
202. Ashaari, M.H., M.G. Osman, and A.H. Mustaffa, *Evaluation of Applications for Nursery School using Machine Learning Approach.* Politeknik & Kolej Komuniti Journal of Life Long Learning, 2021. **5**(1): p. 53–61.
203. BÜYÜKTANIR, B., Ö. Yakar, and A. ALTINEL GİRGİN, *Hate speech framework in social media: A comprehensive review from past to present Sosyal medyada nefret söylemi çerçevesi: Geçmişten günümüze kapsamlı bir derleme.* Journal of the Faculty of Engineering and Architecture of Gazi University, 2024. **40**(1).
204. Mohammed, R., J. Rawashdeh, and M. Abdullah. *Machine learning with oversampling and undersampling techniques: overview study and experimental results*. in *2020 11th international conference on information and communication systems (ICICS)*. 2020. IEEE.
205. Ye, N. and Q. Chen, *An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems.* Quality and reliability engineering international, 2001. **17**(2): p. 105–112.
206. Veerkhare, R., et al. *HRescue: A Modern ML Approach for Employee Attrition Prediction*. in *International Conference on Machine Learning and Big Data Analytics*. 2022. Springer.
207. Hasanin, T. and T. Khoshgoftaar. *The effects of random undersampling with simulated class imbalance for big data*. in *2018 IEEE international conference on information reuse and integration (IRI)*. 2018. IEEE.

ÖZGEÇMİŞ

Adı Soyadı **:** Büşra BÜYÜKTANIR

Öğrenim Durumu

Derece	Bölüm/Program	Üniversite	Mezuniyet Yılı
Lise	Anadolu Lisesi	Hatay Hüseyin Özbuğday Anadolu Lisesi	2010
Lisans	Bilgisayar Mühendisliği	Kırıkkale Üniversitesi	2014
Yüksek Lisans	Bilgisayar Mühendisliği	Marmara Üniversitesi	2022

Akademik ve Mesleki Deneyim

Yıl	Kurum	Görev
2019	Altınbaş Üniversitesi	Araştırma Görevlisi
2021 – Devam	Marmara Üniversitesi	Araştırma Görevlisi

Seçilmiş Bilimsel Yayınlar

1. Büyüktanır, B., Altınkaya, Ş., Karataş Baydoğmuş, G., & Yıldız, K. (2025). *Federated learning in intrusion detection: advancements, applications, and future directions*. Cluster Computing, 28(7), 473.
2. Büyüktanır, B., Baydoğmuş, G. K., Ülkü, E. E., & Yıldız, K. (2026). *Enhancing Employee Retention Prediction Through Federated Learning and Associative Classification using FedCBA: A Case Study on IBM HR Analytics*. Arabian Journal for Science and Engineering.
3. Büyüktanır, B., Yıldız, K., & Baydoğmuş, G. K. (2025). *A systematic mapping study on machine unlearning in federated learning*. 2025 7th International Congress on Human-Computer Interaction, Optimization and Robotic Applications.
4. Öztürk, O., Büyüktanır, B., Baydoğmuş, G. K., & Yıldız, K. (2025). *Differential Privacy in Federated Learning: Mitigating Inference Attacks with Randomized Response*. arXiv preprint arXiv:2509.13987.
5. Demir, A., Kulaksız, A. Y., Büyüktanır, B., Baydoğmuş, G. K., & Yıldız, K. (2026). *Decentralized Disease Prediction: A Federated Learning Perspective*. Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi, 337–353.
6. Kurt, A., Çakır, A., Polat, C. C., Büyüktanır, B., Baydoğmuş, G. K., & Yıldız, K. (2026). *Evaluating SISA-Based Machine Unlearning Across Diverse Modalities: Tabular, Visual and Auditory Data*. Journal of Naval Sciences and Engineering.

7. Büyüktanır, B., Çıplak, Z., Çil, A. E., Yakar, Ö., Adoum, M. B., & Yıldız, K. (2025). *DDoS_FL: Federated learning architecture approach against DDoS attack*. Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 31(6), 1004–1018.
8. Önlü, A. O., Akca, B., Büyüktanır, B., Yıldız, K., & Karataş Baydoğmuş, G. (2025). *An Investigation and Performance Evaluation of Aggregation Algorithms in Federated Learning Architecture*. 2025 Innovations in Intelligent Systems and Applications Conference (ASYU).
9. Tosun, B. E., Yorulmaz, E., Ergül, F. E., Büyüktanır, B., Karataş Baydoğmuş, G., vd. (2025). *Comparative Analysis of Relational Classification Algorithms on Benchmark Datasets*. 2025 10th International Conference on Computer Science and Engineering (UBMK).
10. Büyüktanır, B. (2022). *du-CBA: Data-agnostic and incremental classification-based association rules extraction architecture*. Yüksek Lisans Tezi, Marmara Üniversitesi.