T.C.
NECMETTİN ERBAKAN
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TAMAMEN HOMOMORFİK ŞİFRELENMİŞ VERİLER ÜZERİNDE YAPAY SİNİR AĞLARI PERFORMANS DEĞERLENDİRMESİ

Seyhan AĞLAR

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

NİSAN-2026
KONYA

TEZ KABUL VE ONAYI

Seyhan AĞLAR tarafından hazırlanan "Tamamen Homomorfik Şifrelenmiş Veriler Üzerinde Makine Öğrenimi Performans Değerlendirmesi" adlı tez çalışması 24/04/2026 tarihinde aşağıdaki jüri tarafından oy birliği ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Doç. Dr. Ahmet SINAK	……..
Danışman Dr. Öğr. Üyesi Özlem ERDAŞ ÇİÇEK	……..
Üye Doç. Dr. Ahmet ÖZKIŞ	……..

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun …/…/2026 gün ve …….. sayılı kararıyla onaylanmıştır.

Prof. Dr. Havvanur UÇBEYİAY
FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Seyhan AĞLAR

20/05/2026

ÖZET

YÜKSEK LİSANS TEZİ

TAMAMEN HOMOMORFİK ŞİFRELENMİŞ VERİLER ÜZERİNDE YAPAY SİNİR AĞLARI PERFORMANS DEĞERLENDİRİLMESİ

Seyhan AĞLAR

NECMETTİN ERBAKAN ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman: Dr. Öğr. Üyesi Özlem ERDAŞ ÇİÇEK

2026, 107 Sayfa

Jüri
Doç. Dr. Ahmet SINAK
Doç. Dr. Ahmet ÖZKIŞ
Dr. Öğr. Üyesi Özlem ERDAŞ ÇİÇEK

Bu çalışmada, tamamen homomorfik şifreleme (FHE) kullanılarak şifreli veriler üzerinde makine öğrenmesi yöntemlerinden yapay sinir ağı modellerinin performansı incelenmiş ve şifresiz (açık veri) modellerle karşılaştırılmıştır. FHE, verilerin açığa çıkmadan şifreli biçimde işlenmesini sağlayarak özellikle sağlık, finans ve kişisel veri içeren uygulamalarda gizlilik ve güvenlik açısından kritik bir çözüm sunmaktadır. Veri gizliliğini koruyarak işlem yapmaya imkân veren homomorfik şifreleme yaklaşımı kapsımında özellikle TFHE (Torus Fully Homomorphic Encryption) yöntemi bit temsili, gürültü (noise) oluşumu ve bootstrapping mekanizması ele alınmıştır. Çalışmada farklı veri setleri üzerinde yapay sinir ağı modelleri uygulanmış, katman sayısı (n-layer), nöron sayısı ve epoch gibi hiper parametreler sistematik olarak test edilerek en iyi model yapıları belirlenmiştir. Ayrıca sentetik veriler kullanılarak bu parametrelerin model performansına etkisi analiz edilmiştir. Çalışmada farklı veri setleri üzerinde yapay sinir ağı modelleri uygulanmış, katman sayısı (n-layer), nöron sayısı ve epoch gibi hiper parametreler sistematik olarak test edilerek en iyi model yapıları belirlenmiştir. Ayrıca sentetik veriler kullanılarak bu parametrelerin model performansına etkisi analiz edilmiştir. Deneysel sonuçlar, FHE tabanlı modellerin doğruluk açısından şifresiz modellere oldukça yakın sonuçlar verdiğini göstermiştir. Bununla birlikte, bazı veri setlerinde performans farklılıklarının oluştuğu gözlemlenmiştir. FHE sistemlerinde işlemlerin bit düzeyinde gerçekleştirilmesi ve gürültü birikimi, model davranışını etkileyen önemli faktörlerdir. Bu nedenle bootstrapping mekanizması gürültünün kontrol edilmesinde kritik rol oynamaktadır. Veri boyutu ve özellik sayısının artması, işlem sürelerini doğrudan etkilemektedir. Ayrıca veri ön işleme adımlarının, özellikle normalizasyon tekniklerinin doğru uygulanması model performansı açısından önemli bir gerekliliktir. Sonuç olarak, FHE tabanlı yapay sinir ağı modelleri, veri gizliliği ve güvenliği ile model doğruluğu arasında dengeli bir çözüm sunmakta ve özellikle hassas veri içeren uygulamalarda uygulanabilirliği yüksek bir yöntem olarak değerlendirilmektedir. Bu çalışma, FHE'nin pratik kullanım potansiyelini ve şifreli veri üzerinde yapay sinir ağı modelleri ile güvenli çıkarım yapmanın mümkün olduğunu ortaya koymaktadır.

Anahtar Kelimeler: Concrete-ML, TFHE, Homomorfik Şifreleme, Kriptografi, Makine Öğrenmesi, Yapay Sinir Ağları.

ABSTRACT

MS THESIS

PERFORMANCE EVALUATION OF ARTIFICIAL NEURAL NETWORKS ON FULLY HOMOMORPHIC ENCRYPTED DATA

Seyhan AĞLAR

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF NECMETTİN ERBAKAN UNIVERSITY THE DEGREE OF MASTER OF SCIENCE IN COMPUTER ENGINEERING

Advisor: Asst. Prof. Dr. Özlem ERDAŞ ÇİÇEK

2026, 107 Pages

Jury
Assoc. Prof. Dr. Ahmet SINAK
Assoc. Prof. Dr. Ahmet ÖZKIŞ
Asst. Prof. Dr. Özlem ERDAŞ ÇİÇEK

In this study, the performance of artificial neural network models, one of the machine learning methods, was examined on encrypted data using Fully Homomorphic Encryption (FHE), and the results were compared with unencrypted (plain data) models. FHE provides a critical solution in terms of privacy and security, especially in applications involving sensitive data such as healthcare, finance, and personal information, by enabling data to be processed in encrypted form without being exposed. Within the scope of the homomorphic encryption approach that enables computation while preserving data privacy, the TFHE (Torus Fully Homomorphic Encryption) method was specifically addressed, including bit-level representation, noise generation, and the bootstrapping mechanism. In this study, artificial neural network models were applied on different datasets, and hyperparameters such as the number of layers (n-layer), number of neurons, and epochs were systematically tested to determine the optimal model configurations. In addition, synthetic data were used to analyze the effects of these parameters on model performance. Experimental results showed that FHE-based models achieve accuracy results that are very close to those of plain models. However, it was observed that performance differences may occur in some datasets. In FHE systems, bit-level operations and noise accumulation are important factors affecting model behavior. Therefore, the bootstrapping mechanism plays a critical role in controlling noise. It was also observed that the increase in data size and the number of features directly affects the computational cost and execution time. Furthermore, proper application of data preprocessing steps, especially normalization techniques, is essential for maintaining model performance. As a result, FHE-based artificial neural network models provide a balanced solution between data privacy, security, and model accuracy, and are considered a highly applicable approach, particularly for applications involving sensitive data. This study demonstrates the practical potential of FHE and shows that secure inference on encrypted data using artificial neural networks is feasible.

Keywords: Concrete-ML, Cryptography, TFHE, Homomorphic Encryption, Machine Learning, Neural Networks.

ÖNSÖZ

Bu çalışmanın her aşamasında desteğini ve değerli bilgilerini benden esirgemeyen, yol göstericiliğiyle akademik gelişimime büyük katkılar sunan değerli danışman hocam Dr. Öğr. Üyesi Özlem ERDAŞ ÇİÇEK'e içten teşekkürlerimi sunarım.

Lisans ve yüksek lisans eğitimim boyunca bilgi, deneyim ve desteğiyle akademik gelişmeme önemli katkılar sağlayan, yüksek lisans sürecimde bana inanarak referans olan ve akademik hayatımda ilerlememe destek veren ve tez sürecimde değerli görüşleriyle yol gösteren Doç. Dr. Ahmet SINAK'a içten teşekkürlerimi sunarım.

Bugüne kadar attığım her adımda yanımda olan, zor zamanlarımda bana güç veren, sevgileri ve fedakârlıklarıyla her zaman desteklerini hissettiğim canım anneme ve babama sonsuz teşekkür ederim. Ayrıca hayatım boyunca yanımda olduklarını hissettiğim, destekleriyle bana moral veren değerli kardeşlerime gönülden teşekkür ederim.

Bu süreçte emeği geçen herkese teşekkür eder, çalışmanın akademik camiaya katkı sağlamasını temenni ederim.

Seyhan AĞLAR
KONYA-2026

İÇİNDEKİLER

1. GİRİŞ 1

2. KAYNAK ARAŞTIRMASI 7

2.1. Veri Gizliliği ve Güvenliği 7
2.1.1. Kriptografinin matematiksel temelleri ve yöntemleri 10
2.2. Homomorfik Şifreleme Teknolojileri 13
2.2.1. Kısmi homomorfik şifreleme 14
2.2.2. Seviyeli homomorfik şifreleme 15
2.2.3. Tam homomorfik şifreleme 16
2.2.4. Homomorfik şifreleme sistemlerinde gürültü (Noise) problemi 18
2.2.5. Bootstrapping mekanizması ve hesaplama maliyeti 21
2.3. Homomorfik Şifreleme Şemaları 23
2.3.1. Yaklaşık sayısal hesaplama tabanlı şemalar (CKKS) ve kısıtları 24
2.3.2. Torus tabanlı tam homomorfik şifreleme (TFHE) 28
2.4. Programmable Bootstrapping (PBS) Mekanizması 33
2.5. Yapay Sinir Ağları 36
2.5.1. Yapay sinir ağları yapısı 38
2.5.2. Yapay sinir ağlarında öğrenme 40
2.6. Homomorfik Şifreleme ve Makine Öğrenmesi 40

3. MATERYAL VE YÖNTEM 42

3.1. Kullanılan Veri Setleri 43
3.2. Veri Ön İşleme 50
3.3. Yapay Sinir Ağı Modeli (Scikit-learn) 53
3.3.1. Model mimarisi 55
3.3.2. Eğitim parametreleri 58
3.3.3. Şifresiz eğitim süreci 59
3.4. Concrete-ML ile Şifreli Modelleme 60
3.4.1. Concrete-ML kütüphanesi ve TFHE altyapısı 61
3.4.2. TFHE şeması ve programmable bootstrapping (PBS) 63
3.4.3. Veri nicemleme (Quantization) ve şifreleme süreci 68
3.4.4. Şifreli çıkarım (Encrypted Inference) 69

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA 70

4.1. Performans Değerlendirme Ölçütleri 70
4.2. Şifresiz Model ve FHE Model Karşılaştırması 70
4.3. Normalizasyon ve Ölçekleme Yöntemlerinin Etkisi 85
4.4. Sentetik Veri Üzerinde Model Davranışı 89
4.5. Hesaplama Maliyeti Analizi 91

5. SONUÇLAR VE ÖNERİLER 95

5.1. Sonuçlar 95
5.2. Öneriler 97

6. KAYNAKLAR 98

EKLER 103

ŞEKİLLER LİSTESİ

Şekil 2.1. Genel kriptografi sürecinde anahtar üretimi, şifreleme ve şifre çözme adımları 10
Şekil 2.2. Kriptografiden homomorfik şifrelemeye kavramsal sınıflandırma..... 12
Şekil 2.3. Torus üzerinde düz metin şifreli mesaj alanı..... 20
Şekil 2.4. Torus üzerinde iki şifreli değerin toplanması..... 20
Şekil 2.5. Torus üzerinde NAND kapısının çalışma prensibi..... 21
Şekil 3.1. YSA+FHE deneysel çalışma akış diyagramı 42
Şekil 3.2. Yapay sinir ağı mimarisi (Katman sayısı: 2 veya 3) 54
Şekil 3.3. Homomorfik şifreleme (TFHE) sürecinin işlem adımları..... 67
Şekil 4.1. German Credit veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması..... 71
Şekil 4.2. İris veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması..... 72
Şekil 4.3. Wine veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması..... 73
Şekil 4.4. Parkinson veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması..... 74
Şekil 4.5. Pima Indias Diabetes veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması..... 75
Şekil 4.6. SAHeart veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması..... 76
Şekil 4.7. BUPA veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması..... 77
Şekil 4.8. Heart Disease veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması..... 78
Şekil 4.9. Credit Approval veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması..... 79
Şekil 4.10. Wisconsin Diagnostic Breast Cancer veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması 80
Şekil 4.11. Breast Cancer veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması..... 81
Şekil 4.12. Haberman's Survival veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması..... 82
Şekil 4.13. Sklearn model (şifresiz) ile FHE model (şifreli) üzerinde veri setlerinin doğruluk oranlarının karşılaştırılması..... 84
Şekil 4.14. Z-Score, Min-Max [0,1], Min-Max [-1,1] ve normalizasyonsuz FHE doğruluk sonuçlarının karşılaştırılması..... 89

ÇİZELGELER LİSTESİ

Çizelge 3.1. Kullanılan veri setleri ve ayırıcı özellikleri 49
Çizelge 3.2. Veri setlerine göre yapay sinir ağı model parametreleri 57
Çizelge 3.3. Yapay sinir ağı ve homomorfik şifreleme deneysel süreç adımları algoritması 60
Çizelge 3.4. TFHE şifreleme süreci algoritması 64
Çizelge 4.1. Şifresiz model ve FHE model karşılaştırması 84
Çizelge 4.2. Z-Score standardizasyonu uygulanan veri setlerinde sklearn ve FHE modellerinin doğruluk sonuçlarının karşılaştırılması 86
Çizelge 4.3. Min-Max [0,1] normalizasyonu uygulanmış ve uygulanmamış veri setlerinde sklearn ve FHE modellerinin doğruluk sonuçları 87
Çizelge 4.4. Min-Max [-1,1] normalizasyonu uygulanmış ve uygulanmamış veri setlerinde sklearn ve FHE modellerinin doğruluk sonuçları 88
Çizelge 4.5. Normalizasyon yöntemlerine ve model türüne göre sınıflandırma doğrulukları (%) 88
Çizelge 4.6. Sentetik veri üzerinde model mimarisi ve performans sonuçları 90
Çizelge 4.7. Hesaplama maliyet karşılaştırması 92

SİMGELER VE KISALTMALAR

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

SİMGELER

x_{min}	Özniteliğin minimum değeri
σ_A	Standart sapma
$\mathbb{T}$	Torus uzayı ($\mathbb{R}/\mathbb{Z}$)
$\mathbb{Z}$	Tam sayılar kümesi
$\mathbb{Z}q$	q modülüne göre tam sayılar halkası
R	Polinomlar halkası
R	q modülüne göre polinomlar halkası
a	LWE şifrelemesinde rastgele maske vektörü
b	LWE şifreli metninin ikinci bileşeni
B	GLWE şifreli metninin gövde polinomu
e	Gauss dağılımından örneklenen gürültü terimi
E	GLWE şifrelemesinde gürültü polinomu
f	Aktivasyon fonksiyonu
k	GLWE gizli anahtar vektörünün boyutu / sınıf sayısı
l	GLev ayrıştırma seviye sayısı
m	Düz metin mesajı
M	GLWE şifrelemesinde düz metin polinomu
n	LWE gizli anahtar boyutu / girdi özellik sayısı
N	RLWE halkasındaki polinom derecesi (2'nin kuvveti)
p	Düz metin modülüsü
q	LWE/GLWE modülüsü
s	LWE gizli anahtar vektörü
S	GLWE gizli anahtar polinom vektörü
w	Yapay sinir ağı ağırlık değeri
x	Girdi değeri / özellik
y	Çıktı / tahmin değeri
z	Nöron aktivasyon öncesi toplam girdi
β	GLev/GGSW ayrıştırma tabanı
Δ	Ölçekleme faktörü (q/p)
χ	Gürültü dağılımı (Gauss)

KISALTMALAR

AI	Artificial Intelligence/Yapay Zekâ
ANN-YSA	Artificial Neural Networt/Yapay Sinir Ağı
CKKS	Cheon-Kim-Kim-Song/Cheon-Kim-Kim-Song Şeması
FHE	Fully Homomorphic Encryption/Tam Homomorfik Şifreleme
GLev	Generalized Lev/Genelleştirilmiş Lev Şeması
HE	Homomorphic Encryption/Homomorfik Şifreleme
LUT	Look-Up Table/Arama Tablosu
LWE	Learning With Errors/Hatalı Öğrenme
ML	Machine Learning/Makine Öğrenmesİ
LHE	Leveled Homomorphic Encryption/Seviyeli Homomorfik Şifreleme
PBS	Programmable Bootstrapping/Programlanabilir önyükleme

PHE Partial Homomorphic Encryption/Kısmi Homomorfik Şifreleme
ReLU Rectified Linear Unit/Doğrusal Aktivasyon Fonksiyonu
RLWE Ring Learning With Errors/Halka Tabanlı Hatalı Öğrenme
RSA Rivest-Shamir-Adleman/Rivest-Shamir-Adleman Şeması
SEAL Simple Encrypted Arithmetic Library/Şifreleme Kütüphanesi
SHE Somewhat Homomorphic Encryption/Kısmen Homomorfik Şifreleme
SMPC Secure Multi-Party Computation/Güvenli Çok Taraflı Hesaplama
TenSEAL Tensor Encryption Library/Homomorfik Şifreleme Kütüphanesi
TFHE Fully Homomorphic Encryption Over the Torus/Torus Tabanlı Tam Homomorfik Şifreleme
TLWE Torus Learning With Errors-Torus Hatalı Öğrenme
TRLWE Torus Ring Learning With Errors-Torus Halka Hatalı Öğrenme
UCI UCIrvine Machine Learning Repository- Makine Öğrenmesi Veri Havuzu

1. GİRİŞ

Dijitalleşen dünyada bilgi ve iletişim teknolojilerindeki hızlı gelişmeler sebebiyle kurumlar ve sistemler tarafından üretilen verilerin hacmi her geçen gün çarpıcı bir şekilde artmaktadır. Bu artış, verilerin etkin bir şekilde işlenmesi ve analiz edilmesini hem bilimsel hem de ticari açıdan daha önemli hale getirmiştir. Bu durum, makine öğrenmesi yöntemleri ile büyük veri kümeleri içerisindeki örüntüleri keşfetme, tahmin etme, sınıflandırma, büyük veri kümelerinden anlamlı öngörüler elde etme, karmaşık desenleri tanımlama ve stratejik karar alma süreçlerini desteklemek için güçlü bir araç olarak öne çıkmaktadır. Ancak, bu süreçlerde kullanılan veriler sağlık, finans, savunma sanayi, kamu yönetimi ve e-ticaret gibi alanlarda kişisel, ticari veya hassas bilgiler içermektedir. Bu nedenle veri gizliliği ve güvenliği, yalnızca teknik bir gereksinim değil aynı zamanda yasal ve etik bir zorunluluk olarak karşımıza çıkmaktadır. Özellikle kişisel veriler, finansal bilgiler, kurumsal veriler ve sağlık kayıtları gibi hassas nitelikte olan verilerin korunması yasal ve etik açıdan zorunlu hale gelmiştir. Bu durum, verilerin doğrudan yani açık şekilde, geleneksel makine öğrenmesi teknikleri ile işlenmesinin gizlilik ve güvenlik açısından uygun olmadığını ortaya koymaktadır.

Bu bağlamda, şifrelenmiş veriler üzerinde makine öğrenimi, veri gizliliğini ve güvenliğini koruyarak analitik çalışmaların yapılabilmesini sağlayan yenilikçi bir yaklaşım sunmaktadır. Şifrelenmiş veri işleme teknikleri, veri sahiplerinin mahremiyet kaygılarını giderirken, aynı zamanda verilerin bilgi değerini en üst düzeye çıkarmayı hedeflemektedir. Bu yöntemler sayesinde veriler, şifre çözme işlemine gerek kalmadan analiz edilebilir. Böylece gizlilik ihlallerine karşı önemli bir güvenlik katmanı sağlanmış olur. Özellikle sağlık, finans ve kamu sektörlerinde, bu tür çözümlerin uygulanması büyük bir önem taşımaktadır.

Son yıllarda yapay zekâ, büyük veri ve nesnelerin interneti gibi kavramlarda teknolojideki hızlı gelişim sebebiyle veri üretiminde artışa sebep oluyor. Veri gizliliği ve güvenliği de bu artış sebebiyle çağın temel sorunlarından biri haline gelmektedir. Geleneksel şifreleme yöntemleri ise veri gizliliği ve güvenliğini sağlamada etkili olsa da, şifrelenmiş veriler üzerinde işlemler yapılmasına genellikle izin vermemektedir. Bu nedenle, verilerin şifresi çözülmeden hesaplama yapılabilmesi, gizlilik odaklı veri işleme ve güvenli analiz süreçleri açısından kritik bir gereklilik olarak ortaya çıkmaktadır (Li, 2026).

Şifrelenmiş veriler üzerinde makine öğrenimi uygulamalarında kullanılan teknolojiler arasında son yıllarda homomorfik şifreleme önde gelen yöntemler arasında yer almaktadır. Homomorfik şifreleme, makine öğrenmesi uygulamalarında veri gizliliğini koruyarak analiz yapılmasına imkân tanıyan önemli bir teknoloji olarak öne çıkmaktadır. Geleneksel makine öğrenmesi yöntemleri verilerin açık (plaintext) halde işlenmesini gerektirirken, bu durum hassas verilerin güvenliği açısından ciddi riskler oluşturmaktadır. Homomorfik şifreleme ise veriler şifreli haldeyken aritmetik işlemlerin gerçekleştirilmesine olanak tanıyarak, veri sahiplerinin verilerini ifşa etmeden makine öğrenmesi süreçlerine dahil edebilmesini sağlamaktadır (Liu vd., 2025). Homomorfik şifreleme devrim niteliğinde bir kriptografik yöntem olarak karşımıza çıkmaktadır. Geleneksel şifreleme verilerinin aksine veriler işlenmeden önce şifre çözmeye gerek kalmadan, şifrelenmiş veriler üzerinde doğrudan hesaplamalar yapılmasına olanak tanıyan bir yöntemdir. Bu sebeple güvenilmeyen ortamlarda güvenli verinin işlenmesine, kullanılmasına, istatistiksel analizler yapılmasına, tahmine dayalı modelleme yapılmasına ve yapılan bu çalışmalarda verinin gizli kalmasına imkân tanır (Lee vd., 2025).

Literatür incelendiğinde; 2020 yılında F. Turan ve arkadaşları şifrelenmiş veriler üzerinde homomorfik fonksiyon tahminini hızlandırmak için HEAWS adlı alan merkezli bir yardımcı işlemci sistemi kullanmıştır. Özellikle FV (Fan-Vercauteren) homomorfik şifreleme tekniği kullanılarak, Amazon FPGA'larının büyük boyutundan yararlanılmış ve beş kat daha hızlı enerji tüketimi tahmini yapabilen bir Yapay Sinir Ağı (ANN) tanıtmışlardır (Turan vd., 2020).

2020 yılında A. Mert ve arkadaşları BFV (Brakerski/Fan-Vercauteren) homomorfik şifreleme teknikleri ile Tam Homomorfik Şifreleme (FHE) konusuna bir yaklaşım sunmuştur. İki donanım tasarımı kullanılarak yüksek performanslı polinom çarpanları geliştirilmiş ve şifreleme ile şifre çözme prosedürleri sırasıyla yaklaşık 12 ve 7 kat hızlandırılmıştır (Mert vd., 2020).

2022 yılında A. Kâhya ve arkadaşları Homomorfik Şifreleme ile şifrelenmiş veriler üzerinde makine öğrenimi uygulaması yapmıştır. Büyük veri kümeleri üzerinde makine öğrenme algoritmalarının eğitilmesi ve şifreli verilerle çalışmanın önemini vurgulamıştır. Özellikle tıbbi veriler gibi yüksek gizliliğe sahip verilerin şifrelenmesi ve işlenmesi gerekli görülmüştür. Klasik şifreleme yöntemleri yetersiz kalırken, homomorfik şifreleme (HE) yöntemleri şifreli metin üzerinde polinom işlemlerine izin verdiği için uygun bulunmuştur. Lojistik regresyon, polinom tabanlı bir makine öğrenme algoritması olarak bu çalışmada kullanılmış ve %77,2 başarı oranıyla insanların beş yıl

içinde diyabet teşhisi konulup konulmayacağını tahmin etmiştir. Veri seti CKKS (Cheon Kim Kim Song) HE (Homomorphic Encryption) yöntemiyle şifrelendiğinde, algoritmanın doğru tahmin oranı %76,8 olarak kalmıştır. Bu, homomorfik şifreleme kullanılarak şifrelenmiş veri setleri üzerinde makine öğrenmesi algoritmalarının doğrudan çalıştırılabileceğini göstermektedir (Kâhya, 2022).

Bu çalışma da şifreleme tekniklerinden tamamen homomorfik şifreleme yöntemlerinden TFHE (Torus Fully Homomorphic Encryption) yaklaşımı kullanılarak verilerin şifrelemesi ve şifreli veriler ile yapay sinir ağı modelleri kulanılarak şifrelenmiş veriler üzerinde analiz ve değerlendirmeler yapılmıştır. Elde edilen sonuçlar doğrultusunda yöntemin uygulanabilirliği, performans sonuçları ve şifrelenmemiş hali ile sonuçlarının karşılaştırılması yapılacak de sonuçlar değerlendirilecektir.

Araştırmanın amacı

Makine öğrenimi, verilerden anlamlı bilgileri çıkarmak ve bu bilgileri kullanarak öngörülerde bulunmak için kullanılan bir çeşit yöntemler bütünüdür. Ancak makine öğrenimi, genellikle büyük ve çeşitli veri kümelerinin işlenmesini gerektirir ve bu veriler çoğu zaman hassas, kişisel ya da ticari bilgileri içerebilir. Bu durum, verilerin güvenliğinin sağlanması ve gizliliğin korunması ihtiyacını doğurur. Makine öğrenimi veriler yardımı ile yönlendirildiği için verilerin nicelikleri ve nitelikleri makine öğrenmesi algoritmalarının performanslarını büyük ölçüde etkiler. Ayrıca veri paylaşımı ile ilgili fikri mülkiyet ve gizlilik sorunları sebebiyle veri toplamak zor bir hal alır. Geleneksel makine öğrenimi uygulamaları, verilerin açık bir şekilde erişilebilir olmasını gerektirirken, şifreli veriler üzerinde yapılan makine öğrenimi, bu verilerin güvenli bir şekilde işlenebilmesini sağlar.

Makine öğrenmesi kriptografik sistemleri daha doğru, güvenilir ve sağlıklı hale dönüştürmek için, yapay zekâ tabanlı araçlar yardımıyla düz metin veya şifreli metin verileri ile verilerden anlamlı bilgiler çıkarılması ve veriler arasındaki ilişkilerin belirlenmesi için kullanılabilir. Makine öğrenmesi, öğrenme yapabilmek için algoritmalar ve matematiksel modeller kullanarak, verilerden kalıplar çıkarmayı yani örüntüler elde etmeyi ve bu bilgiler doğrultusunda gelecekteki durumlar için tahminler yapmayı veya kararlar almayı amaçlamaktadır. Yapay zekanın önemli bir alt alanı olan makine öğrenmesi sağladığı verimlilik artışı sayesinde yaygın olarak kullanılmaktadır. Bu sebeple veriler makine öğrenmesi süreçlerinin temel bileşenlerinden biri olarak kritik öneme sahiptir.

Makine öğreniminde kullanılan veriler, modelin eğitimi, doğrulanması ve test edilmesi için üç temel kategoriye ayrılır: eğitim verileri, doğrulama verileri ve test verileri. Eğitim verileri, modelin öğrenmesi gereken kalıpları, ilişkileri ve modelleri belirlemek için kullanılır. Model, bu veri kümesi üzerinden parametrelerini optimize ederek öğrenme sürecini gerçekleştirir. Doğrulama verileri, eğitim sırasında modelin performansını değerlendirmek ve hiper parametre ayarlarını yapmak için kullanılır. Ayrıca, aşırı öğrenme gibi sorunların tespit edilmesine yardımcı olur. Bu veri kümesi, modelin parametrelerini güncellemek için kullanılmaz, yalnızca performansı kontrol etmek amacıyla değerlendirilir. Test verileri, modelin eğitimi tamamlandıktan sonra performansını gerçek dünya senaryolarında ölçmek için kullanılır. Bu veri kümesi, modelin genelleme yeteneğini ve yeni veriler üzerinde nasıl bir performans göstereceğini değerlendirmek için tasarlanmıştır. Bu üç veri kümesinin doğru bir şekilde ayrılması ve kullanılması, modelin başarı oranını ve güvenilirliğini artırmak için kritik öneme sahiptir. Makine öğrenme yöntemleri temel olarak üç kısma ayrılır: denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme (Kâhya, 2022). Bu çalışmada makine öğrenmesi yöntemlerinden yapay sinir ağları kullanılacaktır.

Yapay zekanın son yıllarda hızlı ilerlemesin sebebi derin öğrenme modellerinin başarısından kaynaklanmaktadır. Derin öğrenme modellerinin temel yapı taşını oluşturan yapay sinir ağları, insanın beyin yapısında bulunan biyolojik nöronların çalışma mantığı prensip edinerek geliştirilmiş matematik tabanlı yapılardır. Yapay sinir ağları tahmin yapma, doğal dil işleme, görüntü tanıma gibi çok fazla alanda başarılı sonuçlar elde etmektedir. Yapay sinir ağı girdi katmanı, gizli katman ve çıktı katmanlarından oluşan ve bu katmanlar arasında ağırlıklar (weights) ile sapmalar (biases) aracılığıyla bağlantılar kurulan bir makine öğrenmesi modelidir. Bu yapı sayesinde ağ, eğitim verileri üzerinden öğrenme gerçekleştirerek girdiler ile çıktılar arasındaki karmaşık ve doğrusal olmayan ilişkileri modelleyebilmektedir (Yılmaz ve Şimşek, 2026). Bu çalışmada, söz konusu öğrenme süreci şifrelenmiş veriler üzerinde gerçekleştirilmekte olup, ağın parametreleri olan ağırlık ve sapmaların belirlenmesi homomorfik şifreleme kapsamında ele alınmaktadır. Öğrenme sürecinin başarısı; ağ mimarisi, kullanılan optimizasyon algoritması, kayıp fonksiyonu ve özellikle aktivasyon fonksiyonları gibi temel bileşenlere bağlıdır. Bu bileşenlerin seçimi, şifrelenmiş veri üzerinde gerçekleştirilen işlemlerin doğruluğu ve sistem performansı açısından kritik bir öneme sahiptir. Çalışmada, öğrenme ve çıkarım süreçleri homomorfik şifreleme kapsamında, özellikle TFHE yöntemi kullanılarak şifrelenmiş veriler üzerinde gerçekleştirilmektedir. Bu durum, modelin

parametrelerinin belirlenmesi ve hesaplama süreçlerinin şifreli alan içerisinde yürütülmesini gerektirmektedir. Öğrenme sürecinin başarısı; ağ mimarisi, optimizasyon algoritması, kayıp fonksiyonu ve aktivasyon fonksiyonları gibi temel bileşenlere bağlı olup, bu bileşenlerin seçimi hem doğruluk hem de hesaplama maliyeti açısından kritik bir rol oynamaktadır.

Veri işleme sürecinde Gizlilik ve Güvenliğin Önemi

Günümüzde veri, özellikle dijital çağın en değerli varlıklarından biri haline gelmiştir. Veri toplama, işleme ve analiz süreçleri, birçok sektörde karar verme süreçlerinin temelini oluşturur. Ancak, bu süreçler sırasında verilerin gizliliğini ve güvenliğini sağlamak etik açıdan ve yasalar bakımından bir zorunluluk haline gelmiştir.

Veri işleme sürecinde gizliliğin sağlanması, bireylerin kişisel ve özel bilgilerinin yetkisiz erişimden korunmasını amaçlar. Özellikle sağlık, finans ve kamu sektörlerinde kullanılan veriler, son derece hassas bilgiler içermektedir ve bu bilgilerin korunamaması durumunda bireyler maddi ve manevi ciddi zarar görebilir. Verilerin depolandığı bulutlarla ilgili de çeşitli gizlilik ve güvenlik tehditleri vardır. Bulutta güvenli bir bağlantı oluşturmak veya bir veri tabanını şifrelemek de istemcilere ve bulut hizmeti kaynaklarına yönelik çeşitli saldırılar nedeniyle şifre çözmeye karşı tam koruma sağlamamaktadır (Kadykov vd., 2024).

Günümüzde dijitalleşmenin artmasından dolayı veri miktarında da ciddi oranda artış olmuştur. Verilerin toplanması, depolanması ve işlenip analiz edilmesiyle sık sık karşılaşılmaktadır. Bu durum verilerin gizliliği ve güvenliğinin önemli bir problem haline gelmesine zemin hazırlamıştır. Özellikle veri ihlalleri, zararlı yazılımlar arasında olan oltalama (phishing) saldırıları ve sosyal mühendislik gibi konular tehdit oluşturmakta ve kişisel verilerin yetkisiz erişime maruz kalmasına sebebiyet vermektedir. Bu tarz saldırılar sonucunda kimlik bilgileri, finansal bilgiler ve sağlık verileri gibi hassas veriler kötü amaçla kullanılabilmektedir. Bu nedenle veri gizlilik ve güvenliği yalnızca teknik değil aynı zamanda hukuki düzenlemeler açısından da önemli hale gelmiştir. Gizlilik ihlalleri sadece bireyler için değil, aynı zamanda kurumlar için de yasal ve ekonomik sonuçlara yol açabilir. Türkiye'de kişisel verileri koruma kanunu (KVKK) ve Avrupa'da genel veri koruma yönetmeliği (GDPR) gibi düzenlemeler bu alanda temel çerçeveyi oluşturmuştur. Genel veri koruma yönetmeliği (GDPR) gibi uluslararası düzenlemeler, veri işleme süreçlerinde gizlilik ve güvenlik ilkelerine uyulmasını zorunlu kılmaktadır (Altıntaş ve Barkuş, 2023). Bu nedenle, veri işleme süreçlerinde güvenlik

mekanizmalarının entegre edilmesi büyük önem taşır. Ancak geleneksel güvenlik yöntemleri, verilerin işlenmesi sırasında gizliliğin tam olarak korunmasını sağlayamamaktadır.

Güvenliğin sağlanması, için şifreleme, kimlik doğrulama, erişim kontrolü gibi güvenlik önlemleri, veri işleme süreçlerinin her aşamasında kullanılmalıdır. Ayrıca, gelişmiş güvenlik yöntemleri sayesinde hem verilerin gizliliği hem de bütünlüğü korunarak, yalnızca yetkili tarafların verilere erişimi sağlanır. Makine öğreniminde şifrelemenin sağladığı diğer bir önemli fayda, verilerin işlenmesi sırasında gizliliğin korunmasıdır. Geleneksel makine öğrenimi yöntemlerinde, veriler doğrudan erişilebilir ve analiz edilebilir. Ancak şifrelenmiş verilerle yapılan analizlerde, verilerin açık bir şekilde erişilmesine gerek kalmaz. Şifrelenmiş veriler üzerinde analiz yapılması, homomorfik şifreleme, işbirlikçi makine öğrenimi (federated learning) ve diferansiyel gizlilik gibi tekniklerle mümkün olur. Bu çalışmada, verilerin üçüncü taraflarla paylaşılmadan ve şifre çözülmeden doğrudan işlenmesine olanak tanıması, yüksek düzeyde veri gizliliği sağlaması ve model çıkarımının şifreli veri üzerinde gerçekleştirilebilmesi nedeniyle homomorfik şifreleme yöntemi tercih edilmiştir. Bu yöntemler, verilerin işlenmesi sırasında hem güvenliği hem de gizliliği sağlar. Ayrıca, şifreleme, veri sahiplerinin kontrolünü artırır. Şifreleme teknikleri sayesinde, verinin sahibinin izni olmadan verilere erişilmesi engellenir. Bu, kullanıcıların verilerini daha güvenli bir şekilde paylaşmalarına olanak tanır, çünkü veriler şifreli bir şekilde saklanabilir ve yalnızca belirli koşullarda çözülerek kullanılabilir.

2. KAYNAK ARAŞTIRMASI

Şifreli veriler üzerinde makine öğrenimi yöntemlerinden yapay sinir ağlarını uygulamanın amacı veri gizliliğini ve güvenliğini korumak, sonrasında gizliliği korunmuş yani şifrelenmiş veriler ile model eğitimi yapmak ve bu verilerden etkin bir analiz süreci sağlamaktır. Geleneksel makine öğrenimi yöntemlerinden farklı olarak, verilerin şifreleri çözülmeden ve şifre çözme işlemine ihtiyaç duymadan analiz edilebilmesini hedeflemektedir. Bu teknikler, verilerin şifreli halde kalmasını ve hassas bilgilerin açığa çıkmasını engellerken makine öğrenimi algoritmalarının uygulanmasını sağlar. Şifreli veriler üzerinde makine öğrenimi teknikleri, gizliliği korurken veri analizi ve model eğitimi yapmayı amaçlayan yöntemlerdir.

2.1. Veri Gizliliği ve Güvenliği

Verilerin gizliliği kavramı verinin bu kadar arttığı günümüzde yani büyük veri çağında en önemli ve kıymetli zorluklardan biri olarak görülmektedir. Hassas verileri korumak adına geçmişten günümüze kadar çok fazla şifreleme yöntemleri, güvenlik önlemleri geliştirilmiştir. Kişisel veri kavramı, bireyi doğrudan veya dolaylı olarak tanımlayabilen her türlü kişi hakkında bilgi olarak tanımlanabilmektedir. İsim, kimlik numarası, doğum tarihi gibi temel bilgilerin yanı sıra, bireylerin çevrim içi davranışları, sağlık verileri ve finansal işlemleri de kişisel veri kapsamında değerlendirilmektedir. Bu bağlamda, kişisel verilerin korunması hem bireysel hak ve özgürlüklerin korunması hem de veri güvenliğinin sağlanması açısından kritik bir öneme sahiptir. Veri güvenliği ihlalleri, dijital ortamlarda kişisel verilerin izinsiz erişim, değiştirilme veya ifşa edilmesi gibi risklerin yanında veri setlerinin karşı karşıya kalmasına neden olmaktadır. Kötü amaçlı yazılımlar, oltalama (phishing) saldırıları ve sosyal mühendislik yöntemleri, bu ihlallerin başlıca nedenleri arasında yer almaktadır. Bu nedenle, kişisel verilerin korunması hem teknik önlemler hem de kullanıcı farkındalığı açısından büyük önem taşımaktadır (Altıntaş ve Barkuş, 2023). Bunun yanında yaygın olarak kullanılan veri setlerinin büyük bir kısmı açık ve şifrelenmemiş halde sunulmaktadır. Bu durum, veri üzerinde analiz ve model geliştirme süreçlerini kolaylaştırmakla birlikte, özellikle hassas bilgilerin korunması açısından ciddi güvenlik riskleri oluşturmaktadır. Bu nedenle, verilerin şifreli şekilde işlenmesini sağlayan yöntemler, veri gizliliğinin korunması açısından önemli bir gereklilik haline gelmiştir. Güvenlik için kullanılan planların çoğu, sadece gizli anahtarlara sahip kişilerin şifrelenmiş verilere ulaşabileceği bir varsayıma

dayanmaktadır. Verilerle öğrenme yapılırken makine öğrenmesi yöntemlerinin sık kullanımıyla, etkili ve çözüm odaklı bir modeli eğitmek için verilerin toplanması ardından merkezi bir konuma iletilmesi gerekmektedir. Bu sebeple, özel, hassas ve önemli verilerin sızıntısı durumu her zaman tehlike arz etmektedir. Makine öğrenmesi, bilgi paylaşımı yaparken gizli veri kümelerini veri sızıntısı olmadan nasıl yapacağı önemli bir sorun oluşturmaktadır (Srinivasa Rao vd., 2023).

Verinin hızla artış göstermesiyle büyük verideki teknolojilerinin yanı sıra verilerden anlamlı öngörüler çıkarma, karar verme gibi çok sayıda alanda veri önemli hale gelmiştir. Yapay zekâ alanında görüntü işleme, ses tanıma, doğal dil işleme, makine öğrenmesi teknolojilerinden yapay sinir ağları gibi alanların yanında nesnelerin interneti (IoT) alanında akıllı şehir ve akıllı ev uygulamalarında, bulut bilişim, veri madenciliği, siber güvenlik ve büyük veri gibi alanlarda kritik öneme sahiptir. Bu alanlardaki hassas veriler sebebiyle verilerin gizliliği ve güvenliği kritik öneme sahip hale gelmiştir. Veri gizliliğini sağlamak amacıyla diferansiyel gizlilik yani, güvenli çok taraflı hesaplama, federated learning ve homomorfik şifreleme gibi ileri düzey teknikler geliştirilmiştir (Kabasakaloğlu ve Eyüpoğlu, 2025). Diferansiyel gizlilik, bireysel verilerin korunması için veri üzerine kontrollü gürültü eklenmesini sağlarken, güvenli çok taraflı hesaplama birden fazla tarafın verilerini paylaşmadan ortak hesaplamalar gerçekleştirmesine imkân tanır. Federated learning ise verilerin merkezi bir sistemde toplanmadan, yerel cihazlar üzerinde model eğitimi yapılmasına imkân tanır. Veri anonimleştirme ve veri maskeleme teknikleri ile hassas bilgiler gizlenmekte, klasik kriptografik yöntemler ve erişim kontrol mekanizmaları sayesinde veriye yalnızca yetkili kullanıcıların erişimi sağlanmaktadır. Bu kapsamda, veriler üzerinde şifre çözme işlemi gerçekleştirilmeden hesaplama yapılmasına olanak tanıyan homomorfik şifreleme yöntemi de veri gizliliği alanında önemli bir yaklaşım olarak öne çıkmaktadır. Veriler şifre çözmeden yani gizliliği korunarak eğitilmesi ve sonuçlar çıkarılmasına olanak verir.

Otomotiv sektöründe, özellikle otonom araçların geliştirilmesinde, yolcu güvenliğini sağlamak için yapay zekâ sistemleri bilgisayar korsanlığına karşı güvence altına alınmalıdır. Bu sektörde veri güvenliği, kazalara veya araçların kötüye kullanılmasına yol açabilecek yetkisiz erişimi önlemek için kritik öneme sahiptir. Üretimdeki yapay zekâ hassas endüstriyel verileri ve fikri mülkiyeti içerir. Endüstriyel casusluk ve sabotaja karşı koruma sağlamak için güvenlik önlemlerine ihtiyaç vardır. Üretim yapay zekâ sistemleri genellikle kritik altyapıyı kontrol eder ve bu da güvenliklerini en önemli hale getirir. Eğitimdeki yapay zekâ, öğrenci verilerini ve

öğrenme materyallerini ele alır. Öğrencileri korumak ve eğitim gizliliği yasalarına uymak için eğitim verilerinin güvenliğini ve gizliliğini sağlamak çok önemlidir. Enerji ve kamu hizmetlerinde ise sektördeki yapay zekâ genellikle kritik altyapı verileriyle ilgilenir. Güvenlik zorlukları arasında elektrik veya su temini gibi temel hizmetleri aksatabilecek saldırılara karşı koruma yer alır. Telekom'daki yapay zekâ, müşteri verilerini korumalı ve iletişim ağlarının bütünlüğünü korumalıdır. Güvenlik zorlukları arasında yetkisiz erişime karşı koruma ve iletişim hizmetlerinin güvenilirliğini sağlama yer alır. Tarımdaki yapay zekâ, ürün verimleri, hava desenleri ve çiftlik operasyonlarıyla ilgili verileri işleyebilir. Bu verilerin güvenliğinin sağlanması, çiftçilerin ve gıda tedarik zincirinin mahremiyeti ve ekonomik refahı için hayati önem taşır. Yasal sektördeki yapay zekâ hassas yasal belgeleri ve dava verilerini yönetir. Zorluklar arasında müvekkil gizliliğini korumak, yasal işlemlerin bütünlüğünü sağlamak ve yasal bağlamlarda veri gizliliğini ve güvenliğini yöneten düzenlemelere uymak yer alır. Sigorta sektöründe, yapay zekâ sistemleri poliçe sahipleri ve taleplerle ilgili çok miktarda kişisel ve finansal veriyi işler. Güvenlik endişeleri, dolandırıcılığa karşı koruma, veri işlemenin doğruluğunu sağlama ve müşteri bilgilerini korumak ve sektöre özgü standartlar gibi düzenlemelere uymayı içerir.

Şifreleme kısmında Sağlık Sigortası Taşınabilirliği ve Sorumluluk Yasası (HIPAA) gibi çok sayıda düzenleyici standart vardır. Veri Kaybı Önleme**,** veri güvenliğinde bir diğer önemli unsurdur. Bu uygulamanın niteliği, kazara sızıntıları önlemek, kötü niyetli içeridekileri durdurmak veya bulut tabanlı ortamlarda gizliliği desteklemek olsun, belirli tehdit modeline göre büyük ölçüde değişir. Veri sınıflandırması, hassas veri türlerinin tanımlanmasını, işaretlenmesini ve korunmasını sağlayarak yapay zekâ veri güvenliğinde çok önemlidir. Hassas veriler için doğru koruma önlemlerini sağlayarak veri bütünlüğünü ve gizliliğini korurken ihlal riskini azaltır (Khan, 2019).

Makine öğrenimi gibi veri odaklı teknolojilerde, model eğitimi ve test süreçleri sırasında kullanılan verilerin gizliliğini ve güvenliğini sağlamak kritik öneme sahiptir. Örneğin, şifrelenmiş veri işleme teknikleri ve diferansiyel gizlilik gibi yaklaşımlar, veriler üzerinde analiz yapılmasına olanak tanırken, veri sahiplerinin mahremiyetini korumaya yardımcı olur.

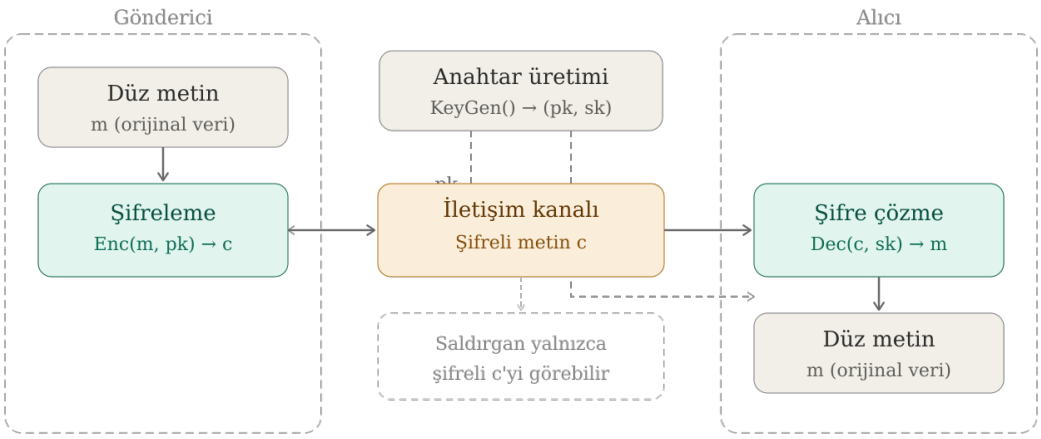
Sonuç olarak, veri işleme sürecinde gizlilik ve güvenliğin sağlanması, yetkisiz erişimleri engellemek adına verinin transfer ve durağanlık süreçlerinde matematiksel olarak korunması yapısal bir zorunluluk haline gelmiştir. Dolayısıyla, modern veri

güvenliği mimarilerinin merkezinde yer alan ve veriyi şifreli hale getirerek güvenli iletimi sağlayan mekanizmaların anlaşılabilmesi için öncelikle bu koruma altyapısını oluşturan temel yöntemlerin incelenmesi gerekmektedir. Bu doğrultuda, veri güvenliğinin sağlanmasında kullanılan temel şifreleme yöntemlerinin ve modern bilgi güvenliği sistemlerinin temelini oluşturan kriptografi kavramından bahsedeceğiz.

2.1.1. Kriptografinin matematiksel temelleri ve yöntemleri

Kriptoloji, bilgi güvenliğinin sağlanmasına yönelik yöntemleri inceleyen bilim dalıdır ve temel olarak kriptografi ile kriptanaliz olmak üzere iki bilim dalından oluşmaktadır. Kriptografi, verilerin yetkisiz erişimlere karşı korunması amacıyla bilgilerin şifrelenmesini sağlayan matematiksel tabanlı yöntemleri kapsarken kriptanaliz, şifrelenmiş verilerin çözülmesine yönelik yöntemleri inceleyerek kriptografik sistemlerin güvenlik açıklarını ortaya çıkarmayı amaçlamaktadır. Günümüzde dijital ortamlarda veri güvenliğinin sağlanabilmesi için özellikle kriptografik yöntemler büyük önem taşımaktadır. Bu nedenle, modern bilgi güvenliği sistemlerinin temelini oluşturan kriptografi kavramının incelenmesi gerekmektedir (Çetin, 2021).

Kriptografi, Yunanca gizli anlamına gelen “kriptos” ve yazı anlamına gelen “graphi” sözcüklerinden türetilmiştir (Çetin, 2021). Temel amacı; verilerin gizliliğinin korunması, güvenli iletişimin sağlanması, veri bütünlüğünün korunması ve yetkisiz erişimlerin engellenmesidir. Kriptografik sistemlerde düz metin (plaintext) olarak ifade edilen veriler, belirli matematiksel algoritmalar ve anahtar yapıları kullanılarak şifreli metin (ciphertext) haline dönüştürülmektedir. Böylece veriler yalnızca yetkili kullanıcılar tarafından tekrar çözülebilir hale gelmektedir.



Şekil 2.1. Genel kriptografi sürecinde anahtar üretimi, şifreleme ve şifre çözme adımları

Şekil 2.1'de açık anahtarlı kriptografik sistemlerin temel çalışma yapısı gösterilmektedir. Sistemde öncelikle anahtar üretim algoritması aracılığıyla açık anahtar (pk) ve gizli anahtar (sk) oluşturulmaktadır. Gönderici tarafında bulunan düz metin (m), alıcının açık anahtarı kullanılarak şifreleme algoritması ile şifrelenmekte ve şifreli metin (c)elde edilmektedir. Şifreleme işlemi denklem (2.1)' de gösterilmiştir (Stallings, 2009).

$$Enc(m, pk) \rightarrow c \tag{2.1}$$

Elde edilen şifreli metin iletişim kanalı üzerinden alıcıya gönderilmektedir. İletim sırasında saldırgan yalnızca şifreli veriye erişebilmekte, verinin orijinal içeriğini görememektedir. Alıcı tarafında ise gizli anahtar kullanılarak şifre çözme işlemi gerçekleştirilmekte ve orijinal düz metin tekrar elde edilmektedir. Şifre çözme işlemi denklem (2.2)' de gösterilmiştir

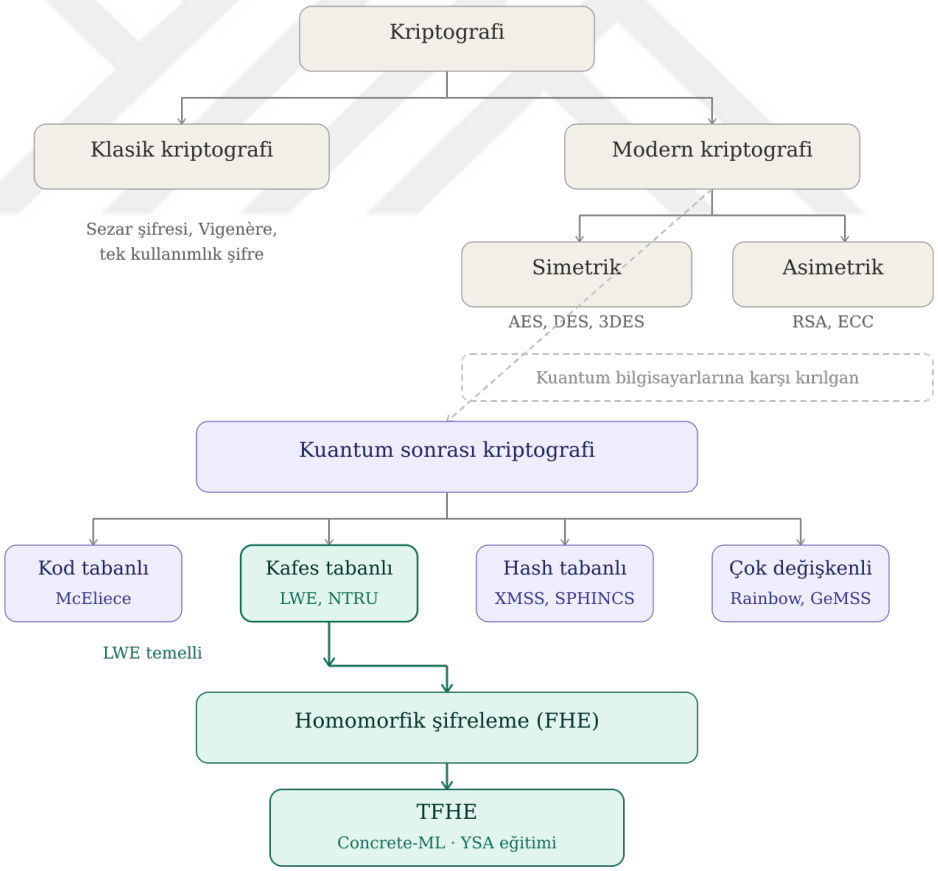
$$Dec(c, sk) \rightarrow m \tag{2.2}$$

Bu yapı sayesinde verilerin yalnızca ilgili gizli anahtara sahip etkili kullanıcı tarafından okunabilmesi sağlanmaktadır (Stallings, 2009).

Geleneksel açık anahtarlı kripto sistemler, günümüz hesaplama sistemlerinde çalışan kriptanaliz yöntemleri ile polinom zamanda çözülemeyen matematiksel problemlere dayandığı için güvenli iletişim sağlamaktadır.

Kuantum bilgisayar teknolojilerindeki gelişmeler, mevcut kriptografik sistemlerin güvenliği açısından yeni tehditleri beraberinde getirmiştir. Kuantum bilgisayar teknolojilerindeki gelişmeler, geleneksel kriptografik sistemlerin güvenliği açısından önemli tehditleri beraberinde getirmiştir. Özellikle RSA, Diffie-Hellman ve eliptik eğri tabanlı kriptosistemlerin güvenliği; tamsayı çarpanlara ayırma ve ayrık logaritma problemlerinin çözümünün zor olması varsayımına dayanmaktadır. Ancak yeterince güçlü kuantum bilgisayarlar üzerinde çalıştırılabilen Shor algoritmasının bu problemlerin polinom zamanda çözülebilmesine olanak sağlayabileceği öngörülmektedir. Bu durum, mevcut açık anahtarlı kripto sistemlerin gelecekte güvenliğini kaybedebileceğini göstermektedir. Bu nedenle hem mevcut hem de gelecekte üretilecek verilerin güvenliğinin korunabilmesi amacıyla kuantum saldırılarına karşı dayanıklı yeni kriptografik sistemlerin geliştirilmesi önem kazanmıştır. Bu kapsamda geliştirilen kuantum sonrası kriptografi yaklaşımları, güvenliklerini kuantum bilgisayarlar tarafından etkin biçimde çözülemeyen matematiksel problemlere dayandırmaktadır. Kuantum

sonrası güvenli kripto sistemler genel olarak kod tabanlı, kafes tabanlı, özet tabanlı ve çok değişkenli polinom sistemleri olmak üzere dört temel sınıfta incelenmektedir. Kod tabanlı sistemler hata düzeltme kodları teorisine dayanırken, özet tabanlı sistemler kriptografik özet fonksiyonlarını temel almaktadır. Çok değişkenli polinom sistemleri ise çok değişkenli polinom denklemlerinin çözümünün zorluğundan yararlanmaktadır. Kafes tabanlı sistemler ise yüksek boyutlu kafes problemlerinin çözümünün hesaplamalı olarak zor olması prensibine dayanmaktadır (Sağıroğlu ve Akleylek, 2021). Özellikle kafes tabanlı kriptografik sistemler kuantum sonrası güvenlik sağlamaları ve homomorfik işlemleri destekleyebilmeleri nedeniyle son yıllarda önemli araştırma alanlarından biri haline gelmiştir. Learning With Errors (LWE) ve Generalized Learning With Errors (GLWE) gibi problemler, kafes tabanlı kriptografinin temel güvenlik yapılarını oluşturmaktadır (Chillotti vd., 2016). Günümüzde birçok modern homomorfik şifreleme sistemi güvenliğini bu problemlerin zorluk varsayımına dayandırmaktadır.



Şekil 2.2. Kriptografiden homomorfik şifrelemeye kavramsal sınıflandırma

2.2. Homomorfik Şifreleme Teknolojileri

Homomorfik şifreleme kavramı ilk kez 1978 yılında Rivest, Adleman, Dertouzos tarafından oluşturulmuştur (Kogos, 2017).Matematiğin temel derslerinden soyut cebirde, homomorfizm, etki alanı ile değer kümesi arasında eşleme yapılırken cebirsel yapıların korunduğu özelliğe denir (Khan, 2019).

Bilgi teknolojileri dünyasında, veriler bulut depolar veya güvenilmeyen bilgisayarlar, servis sağlayıcılar vb. üçüncü taraflarda depolanır ve çeşitli makine öğrenmesi yöntemleriyle işlenir. Bu durum sebebiyle iki büyük endişe ortaya çıkar. Birincisi, eski şifreleme şemaları üçüncü taraftaki verilerin güvenliğini sağlamaz yani şifrelenmemiş verilerin depolanmasına ve tehdit oluşturmasına sebep olur bu durum üçüncü taraflar verileri kötüye kullanmasına sebebiyet verebilir. İkincisi, şifrelenmiş veriler üzerinde herhangi bir işlem yapmak için önce şifresi çözülür, veriler açık ve güvensiz şekilde üçüncü tarafların eline geçebilir ve durum da tehdit oluşturur. Bu endişeler, üçüncü taraflarda şifrelenmiş biçimde depolanan ve şifresi çözülmeden veriler üzerinde hesaplamalar yapılmasına izin veren homomorfik şifrelemenin keşfedilmesine yol açmıştır (Chaudhary vd., 2019). Bu, veri güvenliği ve gizliliği için güçlü bir araç sunar çünkü hassas veriler üzerinde şifrelenmiş haldeyken işlem yapabileceğiniz ve böylece ifşa riskini sınırlayabileceğiniz anlamına gelir.

Homomorfik şifreleme; şifreleme, hesaplama ve şifre çözme adımlarından oluşur. Bu adımları tek tek inceleyelim. Şifreleme, veri sahibi verilerini belirli bir anahtarla şifreler. Bu şifrelenmiş veriler (şifreli metin) daha sonra güvenli olmayan ağlar üzerinden güvenli bir şekilde gönderilebilir, güvenilmeyen bir ortamda saklanabilir çeşitli makine öğrenmesi yöntemleri uygulanabilir, çünkü şifre çözme anahtarı olmadan anlamsız yapılar olarak görünür. Hesaplama ise bir algoritma (bir bulut sunucusu gibi üçüncü bir tarafça kontrol edilebilir) doğrudan bu şifreli metin üzerinde hesaplamalar gerçekleştirir. Homomorfik özellik, şifreli metin üzerindeki işlemlerin düz metin üzerindeki işlemlerle aynı olmasını sağlar. Şifre çözme kısmında ise hesaplamaların sonuçları, hala şifrelenmiş biçimde, veri sahibine geri gönderilir. Veri sahibi, sonucu şifresini çözmek için özel şifre çözme anahtarını kullanır. Şifresi çözülen sonuç, hesaplama orijinal yani şifrelenmemiş veriler üzerinde yapılmış haliyle aynıdır. Homomorfik şifreleme (HE), verilerin şifreli haldeyken matematiksel işlemler yapılmasına olanak tanıyan bir kriptografik şifreleme yöntemidir. Homomorfik Şifreleme, işlemler arasında verileri şifresini çözmek zorunda kalmadan şifrelenmiş veriler üzerinde hesaplamalar yapmaya olanak tanır (T'Jonck vd.,

2022). Bu teknik, verilerin şifresini çözmeden doğrudan şifreli veriler üzerinde işlem yapmayı mümkün kılar. Hiç kimsenin verileri okuyamayacağı veya değiştiremeyeceğinden emin olarak HE, genel bulutlar veya dış taraflar gibi güvenilmeyen ortamlarda bile verileri güvende tutabilir. Homomorfik şifrelemenin temel amacı, verileri şifreleyerek çözmeye gerek duyulmadan, veri şifreli haldeyken hesaplamalar yapılmasına imkân veren kriptografik tekniktir (Kim ve Seok, 2022). Bu durum gizliliğini korurken hesaplama ve analiz yapılmasını sağlamaktır. Farklı türde homomorfik şifreleme şemaları vardır. Bunlar Kısmi homomorfik şifreleme (PHE – Partial Homomorphic Encryption), Seviyeli homomorfik şifreleme (SHE – Leveled Homomorphic Encryption) ve Tam homomorfik şifrelemedir (FHE – Fully Homomorphic Encryption) (T'Jonck vd., 2022). Bu kısımda homomorfik şifrelemenin türlerini inceleyelim.

2.2.1. Kısmi homomorfik şifreleme

Kısmi homomorfik şifreleme (PHE – Partial Homomorphic Encryption) toplama ve çarpma gibi çeşitli matematiksel işlemleri destekler. Bu yöntem yalnızca belirli bir aritmetik işlemi sınırsız kez yapmaya imkân verir (Shaw ve Walde, 2019). Performans açısından daha verimlidir, ancak işlevsellik sınırlıdır.

Kısmi homomorfik şifrelemeye örnek olarak RSA, ElGamal ve Paillier şemaları verilebilir. Paillier şeması toplama işlemi üzerinde homomorfik özellik sağlarken, ElGamal çarpma işlemi açısından homomorfiktir. Bu sistemler, homomorfik şifreleme kavramının ilk pratik örneklerini oluşturmuş ve sonraki gelişmeler için temel teşkil etmiştir (Acar vd., 2019).

Homomorfik şifreleme sayesinde bulut hizmet sağlayıcıları, verilerin şifresini çözmeden bu veriler üzerinde hesaplama yapabilmektedir. Elde edilen sonuçlar da şifreli kalmakta ve yalnızca veri sahibi tarafından çözülebilmektedir. Bu durum, veri gizliliğinin korunmasını sağlarken aynı zamanda hesaplama yapılmasına da imkân tanımaktadır. Homomorfik şifreleme sistemleri genel olarak tamamen homomorfik ve kısmen homomorfik sistemler olarak sınıflandırılmaktadır. Kısmen homomorfik sistemlerde yalnızca belirli işlemler sınırsız sayıda yapılabilirken, bazı sistemlerde birden fazla işlem yapılabilmekte ancak bu işlemlerin sayısı sınırlı olmaktadır. Tamamen homomorfik sistemler ise şifreli veriler üzerinde herhangi bir işlemin gerçekleştirilebilmesine olanak tanımaktadır (Kadykov vd., 2024).

2.2.2. Seviyeli homomorfik şifreleme

Seviyeli homomorfik şifreleme (LHE – Leveled Homomorphic Encryption) belirli bir derinliğe kadar yani, belirli sayıda işlemleri destekler. Performans ve işlevsellik arasında bir denge sunar. Seviyeli homomorfik şifreleme şemaları, şifreli veriler üzerinde hem toplama hem çarpma işlemlerine izin verir. Fakat bu işlemlerin sınırlı sayıda yapılmasına olanak tanır. Bunun temel nedeni şifreleme yapılırken hesaplama adımında, her işlemle birlikte gürültü meydana gelir ve bu artan gürültü (noise) değeridir. Gürültü her işlemle birlikte artış gösterdiği için belirli bir eşik değeri aştığında, şifre çözme işlemi başarısız olur(Acar vd., 2019).

Homomorfik şifreleme, verilerin şifreli haldeyken işlenmesine izin veren bir teknoloji olup, veri gizliliğinin korunması açısından önemlidir. Seviyeli tam homomorfik şifreleme, şifreli veriler üzerinde sınırsız değil, belirli bir derinliğe kadar işlem yapılmasına izin veren bir şifreleme yaklaşımıdır. Bu yöntemde sistem, yalnızca önceden belirlenmiş karmaşıklıktaki hesaplamaları destekleyecek şekilde tasarlanır ve tüm işlemler aynı çözme mekanizması kullanılarak gerçekleştirilir. Çarpımsal derinlik, homomorfik şifreleme şemalarının kapasitesini ve karmaşık hesaplamaları gerçekleştirebilme yeteneğini belirleyen temel kavramlardan biridir. Bu kavram, bir şifreleme sisteminin şifreli veriler üzerinde ardışık olarak gerçekleştirebileceği çarpma işlemlerinin sınırını ifade etmektedir. Bir homomorfik şifreleme sisteminde gerçekleştirilebilecek maksimum çarpımsal derinlik, işlemler sırasında şifreli metin içerisinde oluşan hata (noise) miktarına bağlıdır. Toplama işlemleri hatayı görece sınırlı yani toplamsal olarak artırırken, çarpma işlemleri hatanın çok daha hızlı büyümesine yani hata vektörünün boyutunun birbiriyle çarpılması sonucu çok daha büyümesine neden olmaktadır. Bu hata belirli bir eşik değeri aştığında, şifre çözme işlemi hatalı sonuçlar üretmeye başlar. Bu nedenle, bir devrenin doğru bir şekilde değerlendirilebilmesi için çarpımsal derinliğin kontrol altında tutulması gerekmektedir. Bu noktada iki temel faktör ön plana çıkmaktadır. İlk olarak, homomorfik işlemler sırasında oluşan hata büyümesinin mümkün olduğunca düşük tutulması gerekmektedir. İkinci olarak ise sistemin hata tolerans kapasitesinin yüksek olması, yani belirli bir hata seviyesine kadar doğru sonuç üretebilmesi önemlidir.

Çarpımsal derinlik ve polinom modül derecesi, bu şifreleme türünün hesaplama kapasitesini ve güvenlik seviyesini belirleyen kritik parametrelerdir. Çarpımsal Derinlik, şifrelenmiş veriler üzerinde ardışık olarak kaç çarpma işlemi yapılabileceğini belirten bir

ölçüttür. Her çarpma işlemi, şifrelenmiş verinin içindeki gürültüyü artırır ve belirli bir derinlikten sonra veri güvenilmez hale gelir. Bu parametre, homomorfik şifreleme tekniklerinin hesaplama kapasitesini ve verimliliğini belirler. Yüksek çarpımsal derinlik, daha karmaşık hesaplamaların yapılabilmesini sağlar ancak daha fazla hesaplama gücü ve zaman gerektirir. Polinom Modül Derecesi homomorfik şifrelemelerde kullanılan polinomların derecesini ifade eder. Bu, şifreleme ve şifre çözme işlemlerinde kullanılan polinomların boyutunu ve karmaşıklığını belirler (Gentry, 2009).

Bu derece, şifrelemenin güvenlik seviyesini ve performansını doğrudan etkiler. Yüksek bir polinom modül derecesi, daha güvenli ancak daha fazla hesaplama kaynağı gerektiren bir şifreleme sağlar.

2.2.3. Tam homomorfik şifreleme

Tam homomorfik şifreleme, şifreli veriler üzerinde her türlü işlemin yapılmasına izin veren bir kriptosistemdir. 2009 yılında Gentry tarafından ideal kafeslere dayalı ilk tam homomorfik şifreleme yöntemi geliştirilmiştir. Bu yöntemde gürültü (noise) her işlemle birlikte artar ve belirli bir noktadan sonra şifre çözme işlemi hatalı sonuç verebilir. Bu problemi çözmek için bootstrapping adı verilen bir yöntem kullanılır. Bu yöntem, şifreli veriyi yenileyerek gürültüyü azaltır. Tam homomorfik şifreleme, şifreli veriler üzerinde her türlü işlemin yapılmasına izin veren bir kriptosistemdir. 2009 yılında Gentry tarafından ideal kafeslere dayalı ilk tam homomorfik şifreleme yöntemi geliştirilmiştir. Bu yöntemde gürültü (noise) her işlemle birlikte artar ve belirli bir noktadan sonra şifre çözme işlemi hatalı sonuç verebilir. Bu problemi çözmek için bootstrapping adı verilen bir yöntem kullanılır. Bu yöntem, şifreli veriyi yenileyerek gürültüyü azaltır. Daha sonra bootstrapping kullanılmadan tam homomorfik şifreleme yöntemleri geliştirilmiştir. 2010 yılında bazı yöntemler yalnızca toplama işlemlerine izin verirken, daha sonra hem toplama hem çarpma işlemlerini destekleyen yöntemler geliştirilmiştir. Makine öğrenmesi tabanlı yaklaşımlar da kullanılarak şifreleme desenlerinin öğrenilmesi sağlanmıştır (Gentry, 2009).

Herhangi bir matematiksel işlemi, şifrelenmiş veriler üzerinde gerçekleştirebilir. Performans açısından en maliyetli olanıdır. Tam homomorfik şifreleme sınırsız sayıda işlemin sınırsız sayıda yapılmasına olanak tanır (T'Jonck vd., 2022). Tam homomorfik şifreleme genellikle dört algoritmaya sahiptir: Anahtar oluşturma, şifreleme, şifre çözme ve verimli bir değerlendirme algoritması oluşturmadır. Bu dört algoritmanın süreci şu

şekildedir; ilk olarak anahtar üretilir (Gizli ve açık anahtar), ardından şifreleme yapılır yani düz metni şifrelemek için genel anahtar alır şifreli metni verir, sonraki aşamada değerlendirme yapılır ve son olarak şifre çözülür (T'Jonck vd., 2022).

Tam homomorfik şifrelemeyi diğer şifreleme şemalarından ayıran en önemli kısmı diğer şifreleme şemalarında olduğu gibi sınırlı sayıda değil de sınırsız derinlik içermesine izin vermesidir. Başka bir deyişle bir şifreleme şemasının tam homomorfik olarak adlandırılabilmesi için, yalnızca belirli bir derinlikteki veya sınırlı işlem türlerini destekleyen yapılarla sınırlı kalmaması, bunun yerine her türlü devre üzerinde homomorfik işlemleri gerçekleştirebilmesi gerekmektedir.

İdeal kafes tabanlı şifreleme şemalarında şifreli metin, bir kafes vektörü ile mesajı temsil eden hata bileşeninin toplamı şeklinde ifade edilmektedir. Bu yapı, polinom halkaları üzerinde tanımlanmakta olup şifreli veriler üzerinde yapılan toplama ve çarpma işlemleri, şifresiz veriler üzerinde de aynı işlemlerin gerçekleştirilmesini sağlamaktadır (Acar vd., 2019).

Tamamen homomorfik şifreleme yöntemlerinin altında kafes tabanlı homomorfik şifreleme yöntemleri bulunur. Günümüzde geliştirilen modern homomorfik şifreleme şemalarının büyük bir bölümü, güvenliğini kafes tabanlı kriptografiye dayandırmaktadır. Bu şemalar, özellikle Learning With Errors (LWE) ve Ring-LWE (RLWE) gibi problemlerin hesaplanmasının zorluğuna dayalı olarak güvenlik sağlamaktadır (Regev, 2009). Kafes tabanlı yaklaşımlar, kuantum bilgisayarlara karşı dayanıklı olmaları nedeniyle kuantum sonrası kriptografi bağlamında da önemli bir yere sahiptir. BFV, BGV, CKKS ve TFHE gibi yaygın FHE şemaları bu yaklaşım çerçevesinde geliştirilmiştir (Acar vd., 2019).

Akademi ve endüstri tarafından oluşturulan tam homomorfik şifreleme algoritmalarını uygulayan açık kaynaklı kütüphaneler vardır. Bu kütüphaneler Microsoft SEAL, HElib, PALISADE, TFHE ve TenSEAL dir. Microsoft SEAL; Microsoft tarafından geliştirilmiş olan bu kütüphane, homomorfik şifreleme algoritmalarını kullanarak güvenli hesaplamalar yapmayı sağlar. C++ ve .NET dillerinde kullanılabilir. HElib; IBM Research tarafından geliştirilmiş olan bu kütüphane, homomorfik şifreleme için optimize edilmiştir ve çeşitli homomorfik şifreleme tekniklerini desteklemektedir. C++ dilinde yazılmıştır. PALISADE; bu kütüphane, homomorfik şifreleme, post-quantum kriptografi ve diğer ileri seviye kriptografik işlemler için tasarlanmıştır. C++ dilinde yazılmıştır. TFHE; bu kütüphane, tamamen homomorfik şifreleme için optimize edilmiştir ve özellikle ikili işlemler için uygundur. C++ dilinde yazılmıştır. TenSEAL;

tensor işlemleri üzerinde homomorfik şifreleme uygulamak için tasarlanmış bir kütüphanedir. Python dili ile kullanılabilir (Hamza, 2023). Bu çalışmada TFHE kullanılmıştır.

2.2.4. Homomorfik şifreleme sistemlerinde gürültü (Noise) problemi

Homomorfik şifreleme (HE) sistemlerinin gizliliği ve güvenliği, temelde kafes tabanlı kriptografiye dayanan ve kuantum sonrası kriptografi bağlamında merkezi bir öneme sahip olan "Hata ile Öğrenme (Learning With Errors – LWE)" problemine dayanmaktadır. LWE yaklaşımında, bir mesajın şifrelenmesi sırasında doğrusal denklemlere bilinçli olarak eklenen küçük rastgele hata terimi, şifreli verinin istatistiksel olarak ayırt edilemez hâle gelmesini sağlayarak yetkisiz çözümlemelere karşı güçlü bir güvenlik katmanı oluşturur (Gentry, 2009). Bu hata terimi literatürde *gürültü (noise)* olarak adlandırılmakta olup, HE sistemlerinin güvenliğinin temel bileşenlerinden biri olarak kabul edilmektedir.

Tam Homomorfik Şifreleme (Fully Homomorphic Encryption – FHE) sistemlerinin en temel problemlerinden biri, şifreli uzayda gerçekleştirilen aritmetik işlemler sırasında biriken gürültünün (noise) kontrol altına alınmasıdır. Craig Gentry tarafından 2009 yılında önerilen FHE mimarisi, bu problemi çözerek şifreli veriler üzerinde sınırsız sayıda toplama ve çarpma işleminin gerçekleştirilmesini mümkün kılan ilk teorik yapı olarak literatürde bir dönüm noktası kabul edilmektedir. Gentry'nin yaklaşımı, güvenliğini ideal kafes (ideal lattice) yapılarından almakta ve temelini kafes tabanlı kriptografinin matematiksel olarak zor problemlerine dayandırmaktadır (Gentry, 2009).

Bununla birlikte, gürültü kavramı yalnızca güvenlik sağlayan pasif bir unsur değildir. Şifreli uzayda gerçekleştirilen homomorfik işlemler sırasında gürültü miktarı sabit kalmamakta, aksine yapılan işlemlerin türüne bağlı olarak kademeli biçimde artmaktadır. Özellikle toplama işlemleri gürültüyü doğrusal bir şekilde artırırken, çarpma işlemleri gürültünün çok daha hızlı ve çoğu durumda üstel biçimde büyümesine neden olmaktadır (Gentry, 2009). Bu durum, her bir şifreli metnin taşıyabileceği maksimum gürültü miktarını sınırlayan ve gürültü bütçesi olarak adlandırılan kritik bir kavramın ortaya çıkmasına yol açmaktadır.

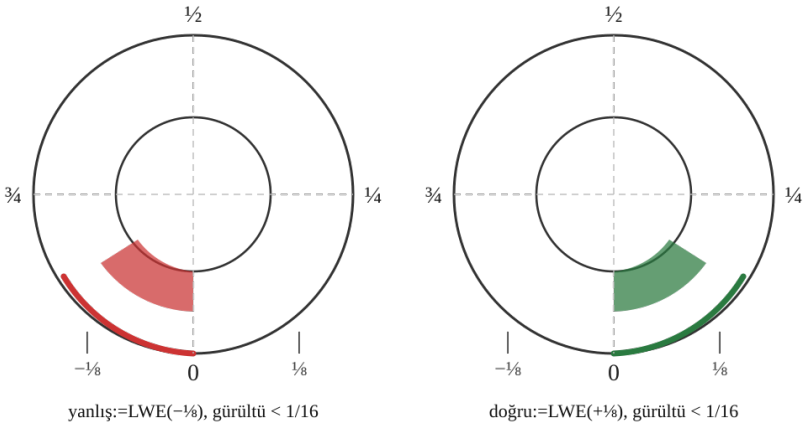
Gürültü bütçesinin aşılması durumunda, şifreli metin içerisinde temsil edilen asıl mesaj gürültü sinyali tarafından baskılanmakta ve şifre çözme işlemi başarısız

olmaktadır. Bu olgu literatürde deşifre hatası olarak tanımlanmakta ve elde edilen sonuçların anlamsız veya hatalı olmasına neden olmaktadır. Özellikle çok katmanlı yapay sinir ağları gibi ardışık ve yoğun çarpma işlemleri içeren modellerde, gürültü birikimi homomorfik hesaplamaların uygulanabilirliğini doğrudan sınırlayan en kritik faktörlerden biri hâline gelmektedir.

Bu tez çalışmasında ele alınan problem bağlamında, gürültü yönetimi yalnızca teorik bir zorluk olarak değil, sistem tasarımını doğrudan etkileyen pratik bir kısıt olarak değerlendirilmiştir. Bu doğrultuda gürültü problemi iki aşamalı bir strateji çerçevesinde ele alınmıştır. İlk aşamada, giriş verileri min–max normalizasyonu kullanılarak (0,1) aralığına ölçeklendirilmiş ve böylece nicemleme (quantization) sürecinde ortaya çıkabilecek sayısal taşmalar ve gereksiz gürültü artışları sınırlandırılmıştır. Bu yaklaşım, şifreli hesaplamalar sırasında kullanılabilir gürültü bütçesinin daha verimli kullanılmasını sağlamıştır.

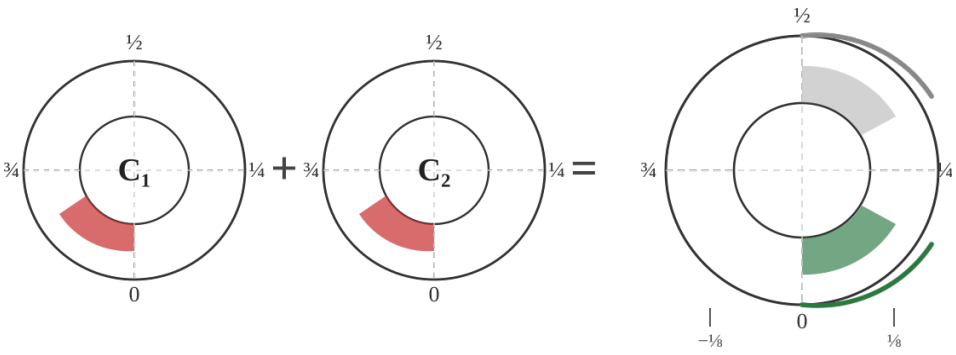
TFHE tabanlı sistemlerde, her işlem adımında uygulanan bootstrapping mekanizması sayesinde şifreli metinler düzenli olarak yeniden şifrelenmekte ve gürültü seviyesi güvenli bir eşik altına indirilmektedir (Chillotti vd., 2016). Bu özellik, klasik homomorfik şifreleme şemalarında görülen sınırlı hesaplama derinliği sorununu büyük ölçüde ortadan kaldırarak, yapay sinir ağları gibi karmaşık ve doğrusal olmayan modellerin şifreli ortamda güvenilir bir şekilde çalıştırılabilmesini mümkün kılmaktadır.

Homomorfik şifreleme, şifreli veriler üzerinde şifre çözmeden işlem yapmayı mümkün kılmaktadır. Torus tabanlı yapılar, özellikle TFHE (Torus Fully Homomorphic Encryption) şemasında, şifreli mesajların temsil edilmesi için kullanılmaktadır. Torus matematiksel olarak $T = \mathbb{R}/\mathbb{Z} \cong \mathbb{R} \bmod 1$ biçiminde ifade edilen dairesel bir yapıdır (Wang vd., 2023). İkili mesaj uzayında bir mesaj yalnızca doğru (true) ya da yanlış (false) değerlerinden birini alabileceğinden, bu iki değer torusun belirli bölgelerine karşılık gelmektedir. Yanlış değer $-1/8$ konumuna, doğru değer ise $+1/8$ konumuna LWE (Learning With Errors) şifrelemesiyle atanmakta, her iki durumda da gürültü 1/16'dan küçük tutulmaktadır. Şifreli mesaj uzayının torus üzerindeki bu gösterimi Şekil 2.3'te verilmiştir.



Şekil 2.3. Torus üzerinde düz metin şifreli mesaj alanı

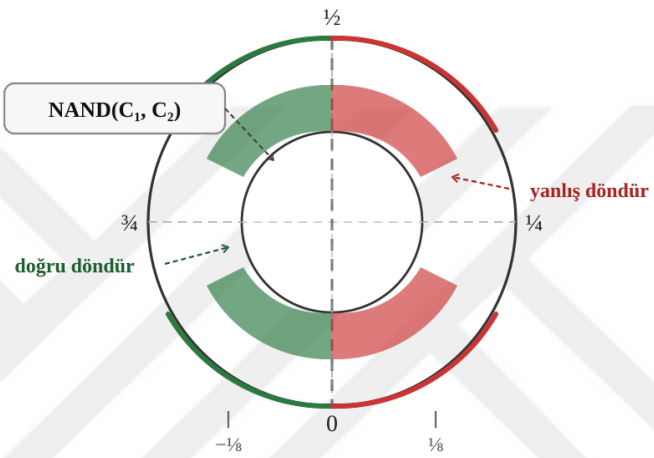
Homomorfik şifrelemede iki şifreli değerin toplanması, torus üzerinde iki ayrı şifreli metnin karşılıklı eklenmesiyle gerçekleştirilir. C_1 ve C_2 olarak adlandırılan iki şifreli metin toplandığında, sonuç torus üzerinde yeni bir konuma düşmektedir. İki torustaki gürültü miktarları da toplandığından, toplam gürültü artmaktadır; ancak bu gürültü belirli bir eşiğin altında kaldığı sürece doğru şifre çözme işlemi yine de gerçekleştirilebilmektedir. Her iki giriş toru da yanlış ise sonuç 3/4 bölgesine, biri doğru diğeri yanlışsa 0 bölgesine ve ikisi de doğruysa 1/4 bölgesine düşmektedir. Bu toplama işleminin torus üzerindeki gösterimi Şekil 2.4'te verilmiştir.



Şekil 2.4. Torus üzerinde iki şifreli değerin toplanması

TFHE kütüphanesinin temelini oluşturan işlem, kapı önyüklemesidir (gate bootstrapping). Her kapı değerlendirmesinin ardından hızlı bir önyükleme gerçekleştirilmekte, bu sayede gürültü her adımda düşük tutulmakta ve işlemler isteğe bağlı derinlikteki devre yapılarına uygulanabilmektedir. Tüm önyükleme kapı devrelerinin doğruluğu, TFHE kütüphanesinde bulunan temel bootsNAND kapısına dayanmaktadır (Wang vd., 2023).

NAND kapısının çalışma prensibi şu şekilde açıklanabilir: C_1 ve C_2 olmak üzere iki giriş verisi varsayılsın. Torus üzerinde dikey bir karar sınırı çizildiğinde, sonucun bu sınırın sol tarafına (doğru bölgesi) ya da sağ tarafına (yanlış bölgesi) düşmesine göre kapının çıkışı belirlenmektedir. Tek bir NAND işleminin ardından sonuç, önyükleme aracılığıyla yenilenmekte ve düşük gürültülü hâle getirilerek sonraki işlemlerde yeniden kullanılabilir kılınmaktadır. Bu prensip, bootsXNOR, bootsAND, bootsOR, bootsMUX ve bootsNOT gibi diğer kapı devrelerine de uygulanmaktadır (Wang vd., 2023). NAND kapısının torus üzerindeki çalışma prensibi Şekil 2.5'te gösterilmiştir.



Şekil 2.5. Torus üzerinde NAND kapısının çalışma prensibi

Şekil 2.5'te şifreli verilerin torus yapısı üzerinde nasıl temsil edildiği ve homomorfik işlemler sırasında nasıl değiştiği gösterilmektedir. Bu nedenle, homomorfik şifreleme sistemlerinde gürültünün artışı önemli bir problem olup, doğru sonuçların elde edilebilmesi için gürültünün sınırlandırılması gerekmektedir.

2.2.5. Bootstrapping mekanizması ve hesaplama maliyeti

Gentry öncesi dönemde geliştirilen homomorfik şifreleme şemaları, sınırlı işlem derinliğine sahip olup yalnızca belirli sayıda aritmetik işleme izin verebilmekteydi. Bu yapıların temel kısıtı, şifreli metin üzerinde yapılan her homomorfik işlemin gürültü miktarını artırmasıdır. Toplama işlemleri gürültüyü doğrusal biçimde artırırken, çarpma işlemleri gürültünün çok daha hızlı ve çoğu durumda üstel olarak büyümesine neden olmaktadır. Gürültü seviyesi belirli bir eşiği aştığında, şifreli metnin doğru şekilde deşifre edilmesi mümkün olmamakta ve sistem hesaplama açısından işlevsiz hâle gelmektedir

(Acar vd., 2018). Bu durum, homomorfik şifreleme sistemlerinde işlem derinliği ile doğruluk arasında doğrudan bir denge problemi ortaya çıkarmaktadır.

Bu temel sınırlamayı aşmak amacıyla Gentry, literatürde "mavi plan" (blueprint) olarak adlandırılan ve Squashing (sıkıştırma**)** ile Bootstrapping (önyükleme) tekniklerini birlikte kullanan bir metodoloji geliştirmiştir (Gentry, 2009). Bu metodoloji, başlangıçta yalnızca sınırlı işlem derinliğine sahip olan Sınırlı Homomorfik Şifreleme (SWHE) yapılarının, teorik olarak sınırsız sayıda homomorfik işlem yapabilen FHE sistemlerine dönüştürülmesini mümkün kılmıştır.

Squashing tekniği, kullanılan şemanın kendi deşifre fonksiyonunu homomorfik olarak çalıştırabilecek yeterliliğe sahip olmaması problemine çözüm getirmektedir. Bu yöntemde, gizli anahtarın çarpımının tersine karşılık gelecek şekilde yapılandırılmış yardımcı bir vektör kümesi tanımlanmakta ve şifreli metin bu vektörler ile birlikte işlenmektedir. Bu sayede deşifre devresinin polinom derecesi azaltılmakta ve sistemin kaldırabileceği işlem derinliği içerisine çekilmektedir. Squashing işlemi sonucunda şema, matematiksel olarak "bootstrappable" yani önyükleme işlemini destekleyebilir hâle getirilmektedir (Acar vd., 2019).

Bootstrapping mekanizması ise, şifreli metin içerisinde biriken gürültüyü ortadan kaldırmak amacıyla uygulanan temel bir yenileme sürecidir. Bu süreçte şifreli metin, açık hâle getirilmeden, tamamen şifreli uzay içerisinde yeniden şifrelenmektedir. Başka bir ifadeyle, sistem kendi deşifre fonksiyonunu homomorfik olarak çalıştırarak gürültüyü temizlemektedir (Gentry, 2009). Bu yaklaşım, klasik kriptografide paradoksal gibi görünse de FHE sistemlerinin temel yeniliğini oluşturmaktadır.

Bootstrapping süreci genel olarak iki farklı anahtar çifti kullanılarak gerçekleştirilmektedir. İlk anahtar çifti ile şifrelenmiş olan ve gürültü seviyesi kritik eşiğe yaklaşmış şifreli metin, gizli anahtarın şifrelenmiş bir kopyası kullanılarak homomorfik deşifre devresinden geçirilmektedir. Bu aşamada veri açık hâle getirilmeden gürültü temizlenmekte, ardından elde edilen sonuç ikinci anahtar çifti kullanılarak yeniden şifrelenmektedir. Böylece ortaya çıkan yeni şifreli metin, orijinal mesajı korumakta ancak önemli ölçüde azaltılmış ve "tazelenmiş" bir gürültü seviyesine sahip olmaktadır (Acar vd., 2018).

Bootstrapping mekanizmasının en önemli avantajı, homomorfik şifreleme sistemlerinde işlem derinliği sınırını teorik olarak ortadan kaldırmasıdır. Ancak bu avantaj, ciddi bir hesaplama maliyeti gerektirmektedir. Gentry'nin ilk önerdiği FHE yapısında bootstrapping işlemi saniyeler hatta dakikalar sürebilmekte ve pratik

uygulamalar açısından ciddi bir performans engeli oluşturmaktaydı (Gentry, 2009). Bu nedenle literatürdeki sonraki çalışmaların büyük bir bölümü, bootstrapping süresini azaltmaya ve bu işlemi daha verimli hâle getirmeye odaklanmıştır (Acar vd., 2018).

Sonuç olarak bootstrapping mekanizması, homomorfik şifreleme sistemlerinde gürültü yönetimini mümkün kılan ve sınırsız hesaplamanın önünü açan temel yapı taşıdır. Her ne kadar yüksek hesaplama maliyeti nedeniyle uzun yıllar boyunca yalnızca teorik bir çözüm olarak değerlendirilmiş olsada, geliştirilen optimizasyon teknikleri ve donanımsal destekler sayesinde günümüzde makine öğrenmesi gibi karmaşık uygulamaların şifreli ortamda çalıştırılabilmesine olanak sağlamaktadır.

Hesaplama maliyetinin temel nedenleri arasında kafes boyutları yer almaktadır. Bu kısımda güvenlik seviyesini artırmak amacıyla kullanılan büyük kafes boyutları, her homomorfik işlemin zaman ve bellek maliyetini doğrudan artırmaktadır. Ardından gürültü yönetiminde ise gürültünün kontrol altında tutulabilmesi için ek matematiksel işlemler gerekmekte, bu da işlem karmaşıklığını yükseltmektedir. Deşifre devresinin karmaşıklığı kısmında ise bootstrapping sırasında homomorfik olarak çalıştırılan deşifre devresi, klasik bir deşifre işlemine kıyasla çok daha fazla işlem içermektedir (Gentry, 2009).

Türkiye'de yapılan akademik çalışmalarda da benzer sonuçlara ulaşılmıştır. Hoşcoşkun (2020), homomorfik şifreleme sistemlerinin özellikle bootstrapping aşamasında yüksek hesaplama yükü nedeniyle gerçek zamanlı uygulamalarda sınırlı kaldığını ve bu durumun donanımsal hızlandırma veya algoritmik iyileştirmelerle aşılabileceğini belirtmiştir (Hoşçoşkun, 2020).

2.3. Homomorfik Şifreleme Şemaları

Homomorfik şifreleme şemaları, şifreli veriler üzerinde doğrudan aritmetik işlemlerin gerçekleştirilebilmesini mümkün kılan kriptografik yapılardır. Bu şemalar güvenliğin sağlanması ve hesaplanabilirliğin korunması açısından kullanılan matematiksel altyapıya göre farklı sınıflara ayrılmaktadır. Günümüzde geliştirilen modern homomorfik şifreleme sistemlerinin büyük bir bölümü, güvenliğini kafes tabanlı kriptografi yaklaşımına dayandırmakta ve özellikle Learning With Errors (LWE) ile Ring-Learning With Errors (RLWE) gibi problemlerin hesaplama açısından çözülemezliğini temel almaktadır (Regev, 2009).

Kafes tabanlı şifreleme yaklaşımları, klasik asal çarpanlara ayırma veya ayrık logaritma problemlerine dayanan geleneksel kriptografik sistemlerin aksine, kuantum bilgisayarlara karşı dayanıklı olmaları nedeniyle kuantum sonrası kriptografi bağlamında da kritik bir öneme sahiptir. Bu bağlamda BFV, BGV, CKKS ve TFHE gibi yaygın olarak kullanılan tamamen homomorfik şifreleme şemaları, farklı hesaplama ihtiyaçlarına ve uygulama senaryolarına yanıt verecek şekilde bu matematiksel temel üzerine inşa edilmiştir (Acar vd., 2018).

Bu tez çalışması kapsamında homomorfik şifreleme şemaları, özellikle makine öğrenmesi uygulamalarında pratik karşılığı bulunan iki temel yaklaşım etrafında ele alınmaktadır. İlk yaklaşım, yaklaşık sayısal hesaplamalara olanak tanıyan ve gerçek sayı aritmetiğini destekleyen CKKS şemasıdır. İkinci yaklaşım ise bit düzeyinde yüksek doğrulukla çalışan, torus tabanlı yapısı ve hızlı bootstrapping mekanizması sayesinde doğrusal olmayan işlemleri etkin biçimde gerçekleştirebilen TFHE şemasıdır. Bu iki şema, matematiksel ve hesaplama modeli açısından farklılıklar göstermekte olup, homomorfik şifreleme ve makine öğrenmesi entegrasyonunun anlaşılması açısından karşılaştırmalı olarak incelenmeye elverişli bir çerçeve sunmaktadır.

Bu bölümün devamında, öncelikle CKKS şemasının yaklaşık hesaplama mantığı, gürültü yönetimi ve sınırlılıkları ele alınacak; ardından TFHE şemasının torus uzayı, LWE tabanlı yapısı ve programmable bootstrapping mekanizması detaylı biçimde açıklanacaktır. Böylece, homomorfik şifreleme şemalarının teorik temelleri ile bu tez kapsamında kullanılan yöntemlerin konumlandırılması sistematik bir biçimde ortaya konulacaktır.

2.3.1. Yaklaşık sayısal hesaplama tabanlı şemalar (CKKS) ve kısıtları

Cheon Kim Kim Song homomorfik şifreleme yöntemi, 2017 yılında Jung Hee Cheon, Andrey Kim, Miran Kim ve Yongsoo Song tarafından geliştirilmiştir. CKKS ismi de yöntemin geliştiricilerinin isimlerinden gelmektedir. HEAAN kütüphanesi olarak bilinmektedir.

Geleneksel tam homomorfik şifreleme (FHE) şemaları tam sayı aritmetiği üzerine kuruludur. Fakat finansal, tıbbi gibi gerçek dünya verilerinin çoğu reel sayılardan oluşur ve bilgisayar sistemlerinde yaklaşık değerler (kayan noktalı sayılar) olarak temsil eder. Var olan yöntemlerin en büyük problemi, şifreli veriler üzerinde yuvarlama (rounding) işlemi yapmanın zorluğudur. İki sayı çarpıldığında anlamlı basamak sayısı artar ve en

önemsiz bitlerin atılması gerekir. Var olan şemalarda bu işlem çok maliyetli "bootstrapping" teknikleri gerektirir ya da şifreli metnin modülünün, işlem derinliği ile beraber üstel olarak büyümesine neden olmakta idi. CKKS yönteminin sunduğu yenilik ile şifreleme sırasında eklenen gürültüyü, şifre çözüldüğünde yok edilmesi gereken bir düşman olarak değil de yaklaşık hesaplamalar sırasında doğal olarak oluşan sayısal hatanın bir parçası olarak görmektedir. Şifre çözme yapısı, şifre çözüldüğünde elde edilen (mesaj + hata) formundadır. Eğer bu hata, mesajın anlamlı basamaklarını bozmayacak kadar küçükse, sonuç başarılı bir yaklaşık değer olarak kabul edilir. Hassasiyet kaybında ise CKKS'nin en önemli özelliklerinden biri olan işlemler sırasında hassasiyet kaybının işlem yapılan devrenin derinliği ile sınırlı olmasıdır. Şifreli işlem sonucu oluşan kayıp, şifresiz kayan noktalı işlemlere kıyas ile en fazla bir bit daha fazla olduğu görülmüştür. CKKS şifreli metnin boyutunu yönetmek ve gürültüyü kontrol altında tutmak için kullanılır. Bu işlem yapılırken öncelikle şifreli metin bir tam sayıya bölünür ve yuvarlanır, eş zamanlı olarak şifreli metin modülü de küçültülür. Sonuç olarak bu işlem kayan noktalı aritmetiğin yuvarlama adımına benzer şekilde mesajın en önemsiz bitlerindeki hatayı temizler. Bu sayede en büyük avantajı, gerekli şifreli metin modülünün boyutunun derinliği ile üstel değil de doğrusal olarak büyümesini sağlamaktadır. Kodlama kısmında verimliliği arttırmak için veriler tek bir şifreli metin içinde paketlenir. Düz metin uzayı, karakteristik sıfır olan bir dairesel halkadır (cyclotomic ring). Karmaşık sayılardan oluşan bir vektör "Karmaşık kanonik Gömme" (Complex Canonical Embedding) haritası kullanılarak bir polinoma dönüştürülür. Bunun avantajı ise bu dönüşüm izometrik bir halka homomorfizmasıdır, yani hataların boyutunu şişirmez ve kodlama sonrası mesajın hassasiyetini korur. CKKS yöntemi, temel toplama ve çarpma işlemlerinin ötesinde, karmaşık analitik fonksiyonların hesaplamasını mümkün kılar. Herhangi bir polinom fonksiyonu, yeniden ölçeklendirme işlemi sayesinde hassasiyet kaybı mimimize edilerek hesaplanır. Ters alma işlemi yapılırken ise bir sayının tersi Newton-Raphson yöntemine benzer yinelemeli bir algoritma ile hesaplanır. Bu sayede Rasyonel fonksiyonların değerlendirilmesi mümkün olur. Üstel ve logistik fonksiyonlarda taylor serisi açılımları kullanılarak hesaplanır. Özellikle lojistik fonksiyon, makine öğrenmesi ve istatistiksel analizlerde hastalık tahmini gibi kritik öneme sahiptir. Şema ve şifreli veriler üzerinde Hızlı Fourier Dönüşümü algoritmasını çalıştırabilir. Paketleme tekniği sayesinde bu işlem yüksek verimlilikle gerçekleştirilir. CKKS yöntemi ile şifreli metnin modülünün büyümesini kontrol altına alarak derin devrelerin hesaplanmasını mümkün kılmıştır ve

şifreli veriler üzerinde veri analizi ve bilimsel hesaplamalar yapmak için pratik ve hızlı bir çözüm sunmaktadır (Cheon vd., 2017).

CKKS yöntemi genel olarak reel sayılar üzerinde yaklaşık hesaplama yapılabilmesi amacı ile Tam Homomorfik Şifreleme şemalarından biridir. Bu yöntemin temel çıkış noktası, makine öğrenmesi ve sinyal işleme tarzı alanlarda gerçekleştirilen hesaplamaların mantığı gereği hata için toleranslı olmasından gelmektedir. CKKS, kesin aritmetik yerine yaklaşık aritmetiği tercih etmektedir ve bu sayede şifreli ortamda reel değerli hesaplamaların pratik bir şekilde yapılmasına olanak sağlar (Kâhya, 2022).

CKKS ile reel sayılar direkt olarak şifrelenmez; bunun yerine, önce belirli bir ölçek faktörü ile çarpılır ardından tamsayıya yakın değerlere dönüştürülür ve polinom yapılar içerisine gömülür. Bu polinomlar, kafes tabanlı kriptografik yapılar kullanılarak şifrelenir. Şifreleme sürecinde mesaja eklenen rastgele gürültü ile sistemin güvenliğini sağlanırken hesaplamalar ilerledikçe oluşan hata terimi oluşur. Bu hata, CKKS'nin tasarımında bilinçli olarak kabul edilen ve kontrol altında tutulan bir bileşendir (Kâhya, 2022).

Şifreli veriler üzerinde toplama işlemleri görece düşük maliyetle gerçekleştirilirken, çarpma işlemleri gürültünün daha hızlı artmasına neden olur. Bu nedenle CKKS, çarpma işlemlerinden sonra yeniden ölçeklendirme adı verilen bir mekanizma kullanır. Yeniden ölçeklendirme, şifreli metnin modülünü küçülterek hem gürültünün kontrol altında tutulmasını sağlar hem de düz metin değerlerinin belirli bir hassasiyet seviyesinde korunmasına imkân tanır. Bu süreçte düz metin değerleri kontrollü bir şekilde yuvarlanır ve böylece hesaplama derinliği boyunca hata birikimi sınırlandırılır.

CKKS yönteminin önemli bir özelliği, çok sayıda reel değerin tek bir şifreli metin içerisinde paketlenebilmesine olanak tanımasıdır. Bu özellik sayesinde vektör ve matris işlemleri, özellikle de makine öğrenmesinde sıkça kullanılan lineer cebir hesaplamaları, yüksek verimlilikle gerçekleştirilebilir. Bu yapı, yapay sinir ağlarının katman hesaplamalarının şifreli ortamda uygulanabilmesini mümkün kılan temel unsurlardan biridir (Kâhya, 2022).

Güvenlik açısından CKKS, Hatalı Anahtar Değişimi ile Halka Öğrenimi (RLWE) problemine dayanmaktadır ve kafes tabanlı kriptografik yöntemler arasında yer alır. Bu yapı, günümüz kriptografik standartlarına göre kuantum saldırılara karşı dayanıklı kabul edilmektedir. Gürültünün belirli bir seviyeyi aşması durumunda sonuçların anlamsız hâle gelmesi riski bulunsa da devre derinliğinin önceden planlanması ve uygun parametrelerin

seçilmesi ile bu durum pratik uygulamalarda kontrol altına alınabilmektedir (Sağıroğlu ve Akleylek, 2021).

CKKS'de gerçekleştirilen tüm aritmetik işlemler yaklaşık sonuçlar üretmektedir. Bu durum, özellikle çarpma işlemleri sonrasında ortaya çıkan yuvarlama ve ölçekleme hatalarının birikmesine yol açabilmektedir. Gürültü seviyesinin kontrolsüz biçimde artması hâlinde, şifreli uzayda elde edilen sonuçlar anlamlılığını yitirebilmekte ve deşifre edilen çıktılar kabul edilemez hata payları içerebilmektedir. Bu nedenle CKKS tabanlı uygulamalarda devre derinliğinin işlem öncesinde dikkatli biçimde planlanması ve parametre seçimlerinin hesaplama gereksinimlerine uygun şekilde yapılması zorunludur (Kâhya, 2022).

Bir diğer önemli kısıt, ölçekleme (scaling) mekanizmasının yönetimidir. CKKS şeması, sayısal hassasiyeti koruyabilmek amacıyla her şifreli değere bir ölçek faktörü atamakta ve çarpma işlemleri sonrasında bu ölçek faktörü üstel olarak büyümektedir. Ölçeklerin yeniden dengelenmesi için uygulanan rescaling işlemi, gürültüyü kısmen azaltmakla birlikte, aynı zamanda veri hassasiyetinde kayıplara neden olabilmektedir. Bu durum, özellikle derin hesaplama zincirlerinde doğruluk–verimlilik dengesinin sağlanmasını zorlaştırmaktadır.

CKKS'nin bir diğer sınırlayıcı yönü, doğrusal olmayan fonksiyonların doğrudan desteklenmemesidir. Aktivasyon fonksiyonları gibi doğrusal olmayan işlemler, genellikle düşük dereceli polinom yaklaşımları ile modellenmek zorunda kalmaktadır. Bu yaklaşım, ek hesaplama maliyeti doğurmakta ve model doğruluğunun sınırlanmasına yol açabilmektedir. Dolayısıyla CKKS, yüksek doğruluk gerektiren bit düzeyindeki işlemlerden ziyade, hata toleransının kabul edilebilir olduğu sürekli değerli hesaplamalar için daha uygun bir yapı sunmaktadır. CKKS şemasının hesaplama ve bellek maliyetleri, klasik şifresiz hesaplamalarla karşılaştırıldığında oldukça yüksektir. Büyük polinom dereceleri, yüksek modül boyutları ve çoklu anahtar yapıları hem işlem süresini hem de bellek kullanımını artırmaktadır. Bu durum, CKKS tabanlı sistemlerin gerçek zamanlı veya kaynak kısıtlı ortamlarda uygulanabilirliğini sınırlayan önemli bir faktör olarak karşımıza çıkmaktadır.

TFHE (Torus Fully Homomorphic Encryption), Chillotti, Gama, Georgieva ve Izabachène tarafından geliştirilen ve literatürde CGGI şeması olarak da bilinen bir tam homomorfik şifreleme sistemidir (Chillotti vd., 2016). TFHE, güvenliğini kafes tabanlı zor problemlerden, özellikle Learning With Errors (LWE) ve halka varyantı Ring LWE (RLWE) probleminden almaktadır. Şemanın diğer FHE sistemlerinden temel farkı,

gürültü azaltma işlemi olan bootstrapping'i son derece hızlı gerçekleştirmesi ve bu işlem sırasında aynı zamanda keyfi bir fonksiyon hesaplayabilmesidir; bu özellik Programlanabilir Bootstrapping (PBS) olarak adlandırılmaktadır (Chillotti, 2022a). Bu kısımda torus tabanlı tam homomorfik şifrelemeden bahsedilecektir.

2.3.2. Torus tabanlı tam homomorfik şifreleme (TFHE)

Torus tabanlı tam homomorfik şifreleme (TFHE), klasik kafes tabanlı şemalardan farklı olarak mesajların tam sayılar veya polinom katsayıları yerine, sürekli bir matematiksel yapı olan Torus üzerinde temsil edilmesine dayanmaktadır. Torus, matematiksel olarak $\mathbb{R}/\mathbb{Z}$ grubu ile ifade edilmekte olup, bu yapı içerisinde tüm gerçek sayılar birim çember üzerinde modüler olarak ele alınmaktadır (Chillotti vd., 2016).

Bu yaklaşımın temel motivasyonu, klasik homomorfik şifreleme şemalarında karşılaşılan taşma (overflow) ve gürültü birikimi problemlerini daha doğal ve kontrollü bir matematiksel çerçeve içerisinde çözmektir. Tam sayı tabanlı şemalarda yapılan ardışık çarpma işlemleri hem mesajın temsil edildiği aralığı hem de gürültü miktarını hızla büyütmekte; bu durum deşifre hatalarına yol açmaktadır. TFHE'de ise mesajlar, [0,1) aralığında Torus üzerinde bir faz (phase) olarak kodlandığından, aritmetik işlemler dairesel bir yapı içerisinde gerçekleştirilmekte ve taşma problemi yapısal olarak ortadan kalkmaktadır (Chillotti vd., 2016).

TFHE tabanlı tam homomorfik şifreleme yaklaşımında sayısal veriler doğrudan işlenmez; bunun yerine sayılar ikili (binary) forma dönüştürülerek her bir bit ayrı ayrı şifrelenir. Bu sayede her bit bağımsız bir şifreli metin olarak temsil edilir. Karşılaştırma işlemi, bu şifreli bitler üzerinde mantık kapıları (XNOR, NOT, MUX vb.) kullanılarak gerçekleştirilir. İşlem, en düşük anlamlı bitten başlanarak ilerler ve her adımda elde edilen ara sonuçlar bir sonraki bit ile birlikte değerlendirilir. Böylece iki sayının hangisinin daha büyük olduğu, veriler çözülmeden belirlenebilir. Bu bit düzeyindeki karşılaştırma mekanizması, bubble sort gibi sıralama algoritmalarında kullanılarak verilerin şifreli hâlde sıralanmasını mümkün kılar. Böylece veri gizliliği korunurken aynı zamanda anlamlı hesaplamalar gerçekleştirilebilir (Wang vd., 2023).

Bu temsil biçimi, özellikle bootstrapping işleminin hem matematiksel olarak sadeleşmesini hem de hesaplama süresinin milisaniye seviyelerine indirgenmesini mümkün kılmıştır. Bu yönüyle TFHE, tam homomorfik şifrelemenin pratik uygulamalara aktarılabilmesinde kritik bir dönüm noktası olarak kabul edilmektedir (Acar vd., 2018).

TFHE şemasının güvenliği ve doğruluğu, kafes tabanlı kriptografinin temel problemlerinden biri olan Learning With Errors (LWE) problemine dayanmaktadır. LWE probleminde temel fikir, doğrusal bir denklem sistemine bilinçli olarak eklenen küçük bir hata (gürültü) terimi sayesinde, gizli anahtarın hesaplanmasının istatistiksel olarak zorlaştırılmasıdır (Regev, 2009).

TFHE, klasik LWE formülasyonunu Torus uzayına uyarlayarak Torus-LWE (TLWE) yapısını kullanır. Bu yaklaşımda hata terimi, mutlak büyüklüğü artan bir değer olarak değil, Torus üzerinde küçük bir açısal sapma olarak modellenmektedir. Bu sayede gürültü, kontrolsüz bir şekilde büyüyen bir unsur olmaktan çıkmakta, belirli bir tolerans aralığında yönetilebilir hâle gelmektedir (Chillotti vd., 2016). Bu matematiksel çerçeve, homomorfik işlemler sırasında oluşan hata birikiminin hem analitik olarak izlenebilmesini hem de bootstrapping mekanizması ile düzenli olarak sıfırlanabilmesini sağlamaktadır.

Torus uzayı, $\mathbb{R}/\mathbb{Z}$ grubu olarak tanımlanmakta olup, tüm gerçek sayıların mod 1 altında ele alınması esasına dayanmaktadır. Bu yapı, sezgisel olarak birim çember ile temsil edilebilir. Bu temsilde her sayı, çember üzerinde bir açıya karşılık gelmektedir. Örneğin, 0 değeri çemberin başlangıç noktasını, 0,5 değeri yarım turu, 0,9 değeri ise çemberin büyük bir bölümünü temsil etmektedir.

Bu yapı içerisinde yapılan aritmetik işlemler modülerdir. Örneğin, klasik reel aritmetikte 0,9 + 0,2 = 1,1 iken, Torus uzayında bu işlem $1 \equiv 0$ kabul edildiğinden sonuç 0,1 olarak elde edilmektedir. Bu modüler süreklilik, homomorfik şifreleme bağlamında taşma problemini doğal bir şekilde ortadan kaldırmaktadır (Özdemir, 2021). Bu kısmı daha net anlaşılmsı için Torus (saat kadranı) üzerinde gerçekleştirilen aritmetik işlemler, klasik doğrusal sayı doğrularından farklı olarak dairesel bir yapıya sahiptir. Örneğin, saat kadranı üzerinde saat 11:00 konumuna 2 saat eklendiğinde sonuç 13:00 olarak değil, tam bir tur tamamlanarak tekrar başlangıç noktasına dönülmesiyle 01:00 (yani 0,1) olarak elde edilir. Bu modüler ve döngüsel yapı, homomorfik şifreleme uygulamalarında karşılaşılan en temel mühendislik problemlerinden biri olan sayısal taşma (overflow) sorununu, verinin kendi tanımlı uzayı içerisinde kalmasını sağlayarak doğal bir biçimde ortadan kaldırmaktadır.

Bu yaklaşımda, hesaplamalar sırasında oluşabilecek değer büyümeleri, doğrusal bir eksende kontrolsüz şekilde ilerlemek yerine dairesel bir uzayda sınırlandırılır. Böylece özellikle ardışık homomorfik işlemler sonucunda ortaya çıkan taşma problemleri, Torus yapısının matematiksel doğası gereği sistemin güvenilirliğini bozmaz hâle gelir.

TFHE sisteminde şifrelenmiş mesajların temsili ve güvenliği sağlayan gürültü (noise) kavramı, saat kadranı üzerindeki akrep benzetmesi ile sezgisel olarak açıklanabilmektedir. Bu analoji, Torus tabanlı şifrelemenin çalışma mantığını kavramayı önemli ölçüde kolaylaştırmaktadır.

Mesaj, orijinal verinin saat kadranı üzerindeki ideal ve net konumu olarak düşünülebilir. Örneğin gizli bir mesajın saat tam 06:00 konumuna, yani Torus üzerinde 0,5 değerine karşılık geldiği varsayılabilir. Bu konum, şifrelenmek istenen gerçek bilginin temsilidir.

Gürültü (hata) ise güvenliği sağlamak amacıyla mesaja bilinçli olarak eklenen küçük bir sapma olarak tanımlanır. Bu durumda saat akrebi tam olarak 06:00 noktasını göstermeyebilir; bunun yerine 06:01 veya 05:59 gibi çok küçük bir açısal titreme sergiler. Bu sapma, saldırganlar için mesajın tam konumunu belirsiz hâle getirirken, yetkili kullanıcı için tolere edilebilir bir hata aralığında kalır.

Deşifre aşamasında sistem, saat akrebi tam olarak 06:00 konumunda olmasa bile, sapma miktarı belirli bir eşik değerin altında kaldığı sürece bu verinin aslında saat 06:00'a karşılık geldiğini doğru biçimde tespit edebilir. Bu tolerans aralığı, şifreleme parametreleri tarafından belirlenen güvenli bir bölgeyi ifade eder.

Ancak şifreli uzayda yapılan homomorfik işlemler arttıkça, bu sapmalar birikerek büyür. Eğer akrep, gürültünün etkisiyle örneğin saat 08:00 konumuna kadar kayarsa, sistem artık mesajın başlangıçta 06:00 mı yoksa 08:00 mı olduğunu ayırt edemez hâle gelir. Bu durum, literatürde gürültü bütçesinin aşılması sonucu verinin bozulması olarak tanımlanmaktadır.

Bu nedenle TFHE sistemlerinde gürültü yönetimi ve özellikle bootstrapping mekanizması, mesajın güvenliğini ve doğruluğunu sürdürebilmek açısından kritik bir rol oynamaktadır. Bu dairesel temsil biçimi, Torus tabanlı şifreleme şemalarında yalnızca sayısal taşma problemini çözmekle kalmaz; aynı zamanda şifreli veriler üzerinde gerçekleştirilen işlemler sırasında oluşan hata ve belirsizliklerin de kontrol edilebilir bir çerçevede kalmasını sağlar. Homomorfik işlemler ilerledikçe, şifreli metinlerin taşıdığı gürültü miktarı artmakta ve bu artış, verinin Torus üzerindeki konumunu giderek daha geniş bir açı aralığına yaymaktadır. Dolayısıyla Torus yapısı, hem aritmetik işlemlerin sürekliliğini garanti eden bir matematiksel zemin hem de gürültü birikiminin etkilerinin sezgisel olarak yorumlanabildiği bir temsil alanı sunmaktadır.

Bu bağlamda, Torus üzerinde gerçekleştirilen hesaplamaların anlaşılabilmesi için, mesajın ve bu mesaja eklenen gürültünün nasıl temsil edildiğinin netleştirilmesi

gerekmektedir. TFHE sistemlerinde bu temsil, çoğunlukla saat kadranı benzetmesi üzerinden açıklanmakta ve gürültü kavramının pratikte nasıl bir etkiye sahip olduğu bu benzetme aracılığıyla daha anlaşılır hâle gelmektedir.

LWE problemi, Regev tarafından tanımlanan ve kafes kriptografisinin temel güvenlik varsayımlarından birini oluşturan zor bir hesaplama problemidir. n boyutlu bir gizli anahtar vektörü $\vec{s} = (s_0, \dots, s_{n-1}) \in \mathbb{Z}^n$ ve q modülüsü verildiğinde, LWE problemi şu iki dağılımı birbirinden ayırt etmeyi gerektirmektedir, rastgele seçilen $(\vec{a}, b) \in \mathbb{Z}_q^{n+1}$çiftleri ile $b = \langle \vec{a}, \vec{s} \rangle + e \pmod q$ biçiminde üretilen çiftler, burada e, standart sapması σ olan Gauss dağılımından örneklenen küçük bir gürültü terimidir (Regev, 2009; Chillotti, 2022a).

Torus tabanlı tamamen homomorfik şifreleme kullanılan cebirsel yapıları tanımlamak için denklem (2.3)'de verilmiştir (Chillotti vd., 2016; Regev, 2009; Chillotti, 2022a).

$$\mathcal{R} = \mathbb{Z}[X]/(X^N + 1) \tag{2.3}$$

N'in 2'nin kuvveti olduğu, derecesi $N - 1$'den küçük tam sayı katsayılı polinomların halkasıdır. Katsayıların q modülüne göre alınması ile elde edilen halka ise aşağıda denklem (2.4)'de tanımlanmıştır (Chillotti, 2022a).

$$\mathcal{R}_q = (\mathbb{Z}/q\mathbb{Z})[X]/(X^N + 1) \tag{2.4}$$

Torus tabanlı tamamen homomorfik şifreleme, LWE ve RLWE şifreli metinlerinin her ikisini de kapsayan GLWE (Generalized LWE) adlı genel bir yapı üzerine inşa edilmiştir. GLWE gizli anahtarı, $\mathcal{R}$ üzerinde k adet rastgele polinomdan oluşan bir vektördür. Denklem (2.5)'de verilmiştir (Chillotti, 2022a).

$$\vec{S} = (S_0, \dots, S_{k-1}) \in \mathcal{R}^k \tag{2.5}$$

$p \leq q$ olmak üzere ölçekleme faktörü $\Delta = q/p$ tanımlandığında, $M \in \mathcal{R}_p$mesajının GLWE ile şifrelenmesi aşağıdaki denklem (2.6)'da verilmiştir (Chillotti, 2022a).

$$(A_0, \dots, A_{k-1}, B) \in GLWE_{\vec{S},\sigma}(\Delta M) \subseteq \mathcal{R}_q^{k+1} \tag{2.6}$$

Burada A_i maskeleri $\mathcal{R}_q$üzerinde düzgün rastgele örneklenir, gövde (body) aşağıda denklem (2.7)'de hesaplanmaktadır (Chillotti, 2022a).

$$B = \sum_{i=0}^{k-1} A_i \cdot S_i + \Delta M + E \in \mathcal{R}_q \quad (2.7)$$

E terimi, katsayıları χ_σ Gauss dağılımından örneklenen gürültü polinomudur. Şifre çözme iki adımda gerçekleşir. İlk olarak (2.8)'de verilen formülle hesaplanır (Chillotti, 2022a).

$$B - \sum_{i=0}^{k-1} A_i \cdot S_i = \Delta M + E \quad (2.8)$$

Bu bağlamda, sonraki aşamadaki ise (2.9)'da yuvarlama işlemi uygulanır (Chillotti, 2022a).

$$M = \lfloor (\Delta M + E)/\Delta \rceil \quad (2.9)$$

Doğru şifre çözme için $| E | < \Delta/2$ koşulunun sağlanması zorunludur; bu koşul ihlal edildiğinde, yani gürültü mesaj ayrımına taştığında şifre çözme hatalı sonuç üretir.

GLWE şeması, $k = n$ve $N = 1$seçildiğinde standart LWE şifrelemesine, $k = 1$ve N2'nin kuvveti seçildiğinde ise RLWE şifrelemesine indirgenir.

Torus tabanlı tamamen homomorfik şifrelemede homomorfik çarpmayı mümkün kılmak için iki ara şifreli metin türü daha tanımlanmıştır. GLev şifreleme, aynı mesajın ℓ farklı ölçekleme faktörüyle şifrelenmiş GLWE şifreli metinlerinden oluşan artıklı bir yapıdır. Denklem (2.10)' da verilmiştir (Chillotti, 2022a).

$$\mathrm{GLev}_{\vec{S},\sigma}^{\beta,\ell}(\mathrm{M}) = \left(\mathrm{GLWE}_{\vec{S},\sigma}\left(\frac{q}{\beta^1}\mathrm{M}\right), \ldots, \mathrm{GLWE}_{\vec{S},\sigma}\left(\frac{q}{\beta^\ell}\mathrm{M}\right)\right) \subseteq \mathcal{R}_q^{\ell\cdot(k+1)} \quad (2.10)$$

Burada β ayrıştırma tabanı ve ℓ seviye sayısıdır. GGSW şifreleme ise bir GLev şifreli metinler vektörüdür. Gizli anahtarın her bir S_i elemanının negatif mesajla çarpımı ile mesajın kendisi GLev olarak şifrelenir. Denklem (2.11)' de verilmiştir (Chillotti, 2022a).

$$\mathrm{GGSW}_{\vec{S},\sigma}^{\beta,\ell}(\mathrm{M}) = \left(\mathrm{GLev}_{\vec{S},\sigma}^{\beta,\ell}(-\mathrm{S}_0\mathrm{M}), \ldots, \mathrm{GLev}_{\vec{S},\sigma}^{\beta,\ell}(-\mathrm{S}_{k-1}\mathrm{M}), \mathrm{GLev}_{\vec{S},\sigma}^{\beta,\ell}(\mathrm{M})\right) \quad (2.11)$$

Homomorfik toplama iki GLWE şifreli metninin bileşen bileşen toplanmasıyla gerçekleştirilir. Denklem (2.12)' de verilmiştir (Chillotti, 2022a).

$$\mathrm{C}(+) = \mathrm{C} + \mathrm{C}' = (\mathrm{A}_0 + \mathrm{A}'_0, \ldots, \mathrm{A}_{k-1} + \mathrm{A}'_{k-1}, \mathrm{B} + \mathrm{B}') \in \mathrm{GLWE}_{\vec{S},\sigma'}(\Delta(\mathrm{M} + \mathrm{M}')) \quad (2.12)$$

Gürültü bu işlemde girdi gürültülerinin toplamı kadar artmaktadır. Büyük sabit ile çarpma doğrudan yapıldığında gürültüyü sabitin büyüklüğüyle orantılı biçimde

artıracağından sorunlu hale gelir. Bu sorunu aşmak için büyük $\gamma \in \mathbb{Z}_q$ sabiti, β tabanında ayrıştırılır. Denklem (2.13)' de verilmiştir (Chillotti, 2022a).

$$\gamma = \gamma_1 \frac{q}{\beta^1} + \gamma_2 \frac{q}{\beta^2} + \cdots + \gamma_\ell \frac{q}{\beta^\ell}, \gamma_j \in \mathbb{Z}_\beta \tag{2.13}$$

Bu küçük γ_j katsayıları, GLev şifreli metninin karşılıklı elemanlarıyla çarpılıp toplanır. Denklem (2.14)' te verilmiştir (Chillotti, 2022a).

$$\langle \mathrm{Decomp}^{\beta,\ell}(\gamma), \bar{C} \rangle = \sum_{j=1}^{\ell} \gamma_j \cdot C_j \in \mathrm{GLWE}_{\vec{S},\sigma'}(\gamma \cdot M) \tag{2.14}$$

Dışsal çarpım (External Product)**,** bir GLWE ile bir GGSW şifreli metninin çarpımıdır ve GLWE şifreli metninin ayrıştırılmasıyla GGSW'nin iç çarpımı alınarak hesaplanır. Denklem (2.15)' de verilmiştir (Chillotti, 2022a).

$$C' = \bar{\bar{C}} \boxdot C = \langle \mathrm{Decomp}^{\beta,\ell}(C), \bar{\bar{C}} \rangle \in \mathrm{GLWE}_{\vec{S},\sigma''}(\Delta M_1 M_2) \tag{2.15}$$

Anahtar değiştirme (Key Switching)**,** bir GLWE şifreli metninin gizli anahtarını $\vec{S}$' den yeni bir $\vec{S}'$anahtarına, mesajı değiştirmeden aktarır. Bu işlem için $\vec{S}$'nin her $S_i$elemanının $\vec{S}'$ altında GLev olarak şifrelendiği bir anahtar değiştirme anahtarı kullanılır. CMux işlemi seçici bir GGSW şifreli metni (b) ve iki GLWE seçeneği (d_0, d_1) alarak homomorfik bir çoklayıcı (multiplexer) değerlendirir. Denklem (2.16)'da verilmiştir (Chillotti, 2022a).

$$\mathrm{CMux}(b, d_0, d_1) = b \cdot (d_1 - d_0) + d_0 = d_b \tag{2.16}$$

Bu işlem, programlanabilir bootstrapping'in temel yapı taşını oluşturmaktadır. Bu kısımda programlanabilir bootsrapping yapısından bahsedelim.

2.4. Programmable Bootstrapping (PBS) Mekanizması

Klasik tam homomorfik şifreleme (FHE) şemalarında bootstrapping mekanizması, temel olarak şifreli veriler üzerinde yapılan işlemler sonucunda biriken gürültüyü temizlemek amacıyla kullanılan yardımcı bir adım olarak ele alınmaktadır. Bu yaklaşımda bootstrapping, hesaplamaya doğrudan katkı sunmayan ancak sistemin çalışabilirliğini sürdüren bir bakım işlemi niteliğindedir. TFHE mimarisi ile birlikte bu anlayış köklü biçimde değişmiş ve bootstrapping işlemi aktif bir hesaplama aracı hâline getirilmiştir. Bu yeni yaklaşım literatürde Programlanabilir Önyükleme (Programmable

Bootstrapping – PBS) olarak adlandırılmaktadır. Programmable Bootstrapping'in temel fikri, gürültü temizleme işlemi sırasında şifreli bir mesaj üzerinde herhangi bir tek değişkenli fonksiyonun doğrudan uygulanabilmesidir. Böylece hem gürültü sıfırlanmakta hem de ek homomorfik işlem derinliğine ihtiyaç duyulmadan fonksiyon hesaplanabilmektedir. Bu özellik, PBS'yi özellikle doğrusal olmayan (non-lineer) fonksiyonların hesaplanmasında TFHE'nin en güçlü bileşenlerinden biri hâline getirmektedir (Chillotti vd., 2016).

Bootstrapping, homomorfik işlemler sırasında büyüyen gürültüyü azaltmak amacıyla şifre çözme fonksiyonunun homomorfik olarak değerlendirilmesi işlemidir. TFHE'de kullanılan bootstrapping mekanizması, yalnızca gürültü azaltımı yapmakla kalmayıp aynı zamanda keyfi fonksiyonları değerlendirebildiği için Programlanabilir Bootstrapping (Programmable Bootstrapping, PBS) olarak adlandırılmaktadır.

PBS üç temel adımdan oluşmaktadır ve bunlar modül değiştirme, kör döndürme ve örnek çıkarma işlemleridir. İlk aşama olan modül değiştirme işleminde giriş LWE şifreli metninin modülüsü q'dan $2N$'e indirgenmektedir Denklem (2.17)'de verilmiştir (Chillotti, 2022b).

$$\tilde{a}_i = \lfloor \frac{2N \cdot a_i}{q} \rceil \in \mathbb{Z}_{2N} \tag{2.17}$$

Bu işlem sırasında mesaj bilgisi korunurken gürültü ile mesaj arasındaki göreli mesafe azaltılmaktadır. PBS'nin temelini oluşturan kör döndürme (Blind Rotation) aşamasında ise LUT (Look-Up Table) içeren bir polinom ve bootstrapping anahtarı kullanılarak polinom şifreli biçimde döndürülmektedir Denklem (2.18)'de verilmiştir (Chillotti, 2022b).

$$V_n = V \cdot X^{-\tilde{b}} \cdot \prod_{i=0}^{n-1} X^{\tilde{a}_i s_i} = V \cdot X^{-(\tilde{\Delta} m + \tilde{e})} \tag{2.18}$$

Bu işlem sonucunda elde edilen GLWE şifreli metninin sabit katsayısında fonksiyon sonucu yer almaktadır. Son aşama olan örnek çıkarma (Sample Extraction) işleminde ise GLWE şifreli metninin sabit katsayısı bir LWE şifreli metni olarak çıkarılmaktadır. PBS'nin en önemli özelliği, LUT içerisinde herhangi bir fonksiyonun kodlanabilmesidir. Böylece gürültü azaltımı ve fonksiyon değerlendirmesi tek işlemde

birleştirilmektedir. Örneğin yapay sinir ağlarında kullanılan ReLU aktivasyon fonksiyonu aşağıdaki şekilde ifade edilebilmektedir Denklem (2.19)'da verilmiştir (Chillotti, 2022b).

$$f_m = \max(0, m) \tag{2.19}$$

Bu özellik sayesinde TFHE, yapay sinir ağlarının şifreli çıkarım süreçlerinde kullanılabilmekte ve doğrusal olmayan aktivasyon fonksiyonlarının şifreli veriler üzerinde değerlendirilebilmesine olanak sağlamaktadır.

Programlanabilir Önyükleme (Programmable Bootstrapping – PBS) mekanizmasının matematiksel işleyişi, Torus uzayı üzerinde gerçekleştirilen kör döndürme (blind rotation) işlemi etrafında şekillenmektedir. Bu işlem, TFHE şemasının hem güvenliğini hem de hesaplama yeteneğini mümkün kılan temel yapı taşlarından biridir. PBS sürecinin girdisi, Torus üzerinde bir açıyı temsil eden ve gürültü seviyesi artmış bir TLWE şifreli metnidir. Bu şifreli metnin, Torus uzayında kodlanmış bir x mesajını içerdiği varsayılmaktadır. Şifreli metnin taşıdığı faz bilgisi, doğrudan okunamamakla birlikte, homomorfik işlemler aracılığıyla dolaylı biçimde kullanılabilmektedir.

PBS işlemi sırasında gerçekleştirilen kör döndürme adımında, sistemde önceden tanımlanmış bir test polinomu, şifreli metnin içerdiği gizli değer x kadar döndürülür. Bu döndürme işlemi, gizli anahtarın açık hâli kullanılmadan, anahtarın şifreli kopyaları üzerinden gerçekleştirilir. Bu nedenle sunucu, döndürme miktarını ve dolayısıyla mesajın gerçek değerini hiçbir aşamada öğrenmez. Bu özellik, PBS'nin temel güvenlik garantilerinden birini oluşturmaktadır (Chillotti vd., 2016).

Hesaplanmak istenen $f(x)$ fonksiyonu, PBS işlemi başlamadan önce test polinomunun katsayılarına kodlanır. Bu yaklaşım, matematiksel olarak (Look-Up Table – LUT) mantığına karşılık gelmektedir. Polinomun her bir konumu, fonksiyonun belirli bir giriş değeri için ürettiği çıktıyı temsil edecek şekilde düzenlenir. Kör döndürme işlemi tamamlandığında, polinomun sabit terimi doğrudan $f(x)$ değerini içerecek biçimde hizalanmış olur.

PBS işleminin çıktısı hem gürültüsü temizlenmiş hem de tanımlanan fonksiyon uygulanmış yeni bir TLWE şifreli metnidir. Böylece klasik bootstrapping işlemlerinde olduğu gibi yalnızca gürültü sıfırlanmakla kalmaz; aynı zamanda şifreli veri üzerinde aktif bir hesaplama gerçekleştirilmiş olur (Acar vd., 2019).

2.5. Yapay Sinir Ağları

Yapay sinir ağları (YSA), insan beyninin öğrenme yolunu taklit ederek beynin öğrenme, hatırlama, genelleme yapma yolu ile topladığı verilerden yeni veri üretebilme gibi temel işlevlerin gerçekleştirildiği bilgisayar yazılımlarıdır (Öztürk, 2018).

Yapay sinir ağları, insan beyninin bilgi işleme biçiminden esinlenilerek geliştirilmiş ve veriler arasındaki karmaşık ve doğrusal olmayan ilişkileri öğrenebilen makine öğrenmesi modelleridir. Bu modeller, birbirine bağlı çok sayıda yapay nörondan oluşur ve her bir nöron, kendisine gelen girdileri belirli ağırlıklar ile çarparak toplar, elde edilen sonuca bir yanlılık (bias) ekler ve bu değeri bir aktivasyon fonksiyonundan geçirerek çıktı üretir. Yapay sinir ağlarının temel amacı, verilen örnekler üzerinden öğrenerek daha önce karşılaşılmamış veriler için doğru tahminlerde bulunabilmektir.

Bir yapay sinir ağı genellikle giriş katmanı, bir veya daha fazla gizli katman ve çıkış katmanından oluşur. Giriş katmanı, modele verilen ham verileri temsil ederken, gizli katmanlar bu veriler üzerinde özellik çıkarımı ve dönüşüm işlemlerini gerçekleştirir. Çıkış katmanı ise problemin türüne bağlı olarak sınıflandırma, regresyon veya olasılık tahmini gibi sonuçlar üretir. Gizli katman sayısı ve her katmandaki nöron sayısı, ağın öğrenme kapasitesini ve karmaşıklığını belirleyen önemli parametrelerdir.

Yapay sinir ağlarının öğrenme süreci, eğitim verileri kullanılarak ağ ağırlıklarının iteratif olarak güncellenmesine dayanır. Bu süreçte, modelin ürettiği çıktılar ile gerçek değerler arasındaki fark bir kayıp (loss) fonksiyonu ile ölçülür ve bu hata, geri yayılım (backpropagation) algoritması kullanılarak ağın katmanları boyunca geriye doğru dağıtılır. Geri yayılım sırasında, her bir ağırlığın hataya olan katkısı hesaplanır ve ağırlıklar gradyan inişi gibi optimizasyon yöntemleri ile güncellenir. Böylece ağ, her iterasyonda hatayı azaltacak şekilde öğrenme sürecini sürdürür.

TFHE'nin yapay sinir ağlarında kullanılabilmesinin temel nedeni, bir nöronun hesaplama yapısının homomorfik işlemler ile temsil edilebilmesidir. Bir yapay sinir ağında nöron çıktısı aşağıdaki şekilde ifade edilmektedir. Denklem (2.20)' de verilmiştir (Chillotti, 2022c).

$$y = f(\sum_{i=1}^{n} w_i\, x_i + b) \tag{2.20}$$

Burada $x_i$giriş değerlerini, w_iağırlıkları, $b$bias değerini ve $f$aktivasyon fonksiyonunu temsil etmektedir. Yapay sinir ağlarında ilk aşamada girişler ile ağırlıklar

arasında doğrusal birleşim hesaplanmakta, ardından elde edilen sonuç aktivasyon fonksiyonundan geçirilmektedir. TFHE'de doğrusal işlemler, LWE şifreli metinleri üzerinde gerçekleştirilen homomorfik toplama ve sabitlerle çarpma işlemleri aracılığıyla uygulanabilmektedir. Buna göre doğrusal birleşim işlemi şifreli veri üzerinde aşağıdaki şekilde gerçekleştirilebilmektedir. Denklem (2.21)' de verilmiştir (Chillotti, 2022c).

$$\mathrm{Enc}(z) = \sum_{i=1}^{n} w_i \cdot \mathrm{Enc}(x_i) + b \tag{2.21}$$

Ancak aktivasyon fonksiyonları doğrusal olmayan yapılar olduğundan, homomorfik şifreleme sistemlerinde doğrudan değerlendirilmeleri mümkün değildir. TFHE'de bu problem Programlanabilir Bootstrapping (PBS) mekanizması ile çözülmektedir. PBS sırasında aktivasyon fonksiyonu bir LUT (Look-Up Table) içerisine kodlanmakta ve bootstrapping işlemi sırasında fonksiyon değerlendirmesi gerçekleştirilmektedir. Bu çalışmada kullanılan ReLU aktivasyon fonksiyonu aşağıdaki şekilde tanımlanmaktadır. Denklem (2.22)' de verilmiştir (Chillotti, 2022d).

$$f(z) = \max(0, z) \tag{2.22}$$

PBS işlemi sonucunda aktivasyon fonksiyonu şifreli veri üzerinde aşağıdaki şekilde değerlendirilmektedir. Denklem (2.20)' de verilmiştir (Chillotti, 2022d).

$$\mathrm{PBS}_{ReLU}(\mathrm{Enc}(z)) = \mathrm{Enc}(\max(0, z)) \tag{2.23}$$

Bu yapı sayesinde hem gürültü azaltımı yapılmakta hem de aktivasyon fonksiyonu şifreli veri üzerinde uygulanabilmektedir. Böylece her nöronun çıktısı şifreli olarak elde edilmekte ve bir sonraki katmana giriş olarak aktarılabilmektedir. TFHE'nin bu özelliği, yapay sinir ağlarının şifreli çıkarım süreçlerinde kullanılabilmesine olanak sağlamaktadır.

Yapay sinir ağlarının en önemli özelliklerinden biri, doğrusal olmayan aktivasyon fonksiyonları sayesinde karmaşık örüntüleri modellemesidir. Sigmoid, tanh ve ReLU gibi aktivasyon fonksiyonları, ağın basit doğrusal modellerin ötesine geçmesini sağlar. Bu sayede yapay sinir ağları, görüntü işleme, ses tanıma, doğal dil işleme ve tıbbi veri analizi gibi pek çok alanda başarılı sonuçlar üretmektedir.

Sonuç olarak yapay sinir ağları, büyük ve karmaşık veri kümelerinden anlamlı bilgi çıkarabilen, esnek ve güçlü öğrenme modelleridir. Öğrenme yetenekleri, katmanlı yapıları ve doğrusal olmayan hesaplamaları sayesinde klasik istatistiksel yöntemlerin

yetersiz kaldığı problemlerde etkili çözümler sunarlar. Bu özellikleri nedeniyle, günümüzde makine öğrenmesi ve yapay zekâ alanlarının temel yapı taşlarından biri olarak kabul edilmektedir.

Yapay sinir ağı modelleri tek katmanlı algılayıcılar, çok katmanlı algılayıcılar, ileri beslemeli yapay sinir ağları ve geri beslemeli yapay sinir ağları olarak dört gurupta incelenebilir.

2.5.1. Yapay sinir ağları yapısı

Yapay sinir ağları, biyolojik sinir sisteminden esinlenilerek geliştirilen ve giriş–çıkış ilişkilerini öğrenebilen matematiksel modellerdir. Bu ağlar, temel olarak yapay nöronlardan oluşur ve bu nöronlar katmanlı bir yapı içerisinde birbirlerine bağlanarak çalışır. Yapay sinir ağlarının yapısı; yapay nöron modeli, katmanlı ağ mimarisi, ileri beslemeli ağ yapısı ve aktivasyon fonksiyonları olmak üzere dört temel bileşen altında incelenmektedir (Haykin, 2009).

2.5.1.1. Yapay nöron modeli

Yapay sinir ağlarının en küçük yapı taşı yapay nörondur. Yapay nöron, biyolojik nöronların çalışma prensibinden esinlenilerek modellenmiştir. Bir yapay nöron, kendisine gelen giriş sinyallerini belirli ağırlık katsayıları ile çarpar ve bu değerleri toplar. Elde edilen toplam değere bir bias (eşik) terimi eklenerek sonuç bir aktivasyon fonksiyonuna iletilir ve nöron çıktısı üretilir.

Matematiksel olarak yapay nöron modeli aşağıdaki şekilde ifade edilir. Denklem (2.24)'de verilmiştir.

$$y = f(\sum_{i=1}^{n} w_i\, x_i + b) \tag{2.24}$$

Burada x_i giriş değerlerini, w_iağırlıkları, b bias terimini ve $f(\cdot)$ aktivasyon fonksiyonunu temsil etmektedir. Yapay nöron modeli, sinir ağlarının öğrenme ve genelleme yeteneğinin temelini oluşturmaktadır (Haykin, 2009).

Katmanlı ağ yapısı Yapay sinir ağları genellikle katmanlı bir mimariye sahiptir. Bu mimari; giriş katmanı, bir veya daha fazla gizli katman ve çıkış katmanından oluşur. Giriş katmanı, dış ortamdan alınan verilerin ağa aktarıldığı katmandır ve bu katmanda herhangi bir hesaplama işlemi yapılmaz.

Gizli katmanlar, giriş verilerinin işlenerek daha soyut temsillere dönüştürüldüğü katmanlardır. Öğrenme süreci esas olarak bu katmanlarda gerçekleşir. Çıkış katmanı ise ağın nihai çıktısını ürettiği katmandır ve problem türüne göre sınıflandırma veya regresyon sonuçları üretir.

Katman sayısı ve her katmandaki nöron sayısı, ağın ifade gücünü doğrudan etkilemektedir. Ancak aşırı derin ağlar, hesaplama maliyetini artırmakta ve aşırı öğrenme (overfitting) riskini beraberinde getirmektedir(Haykin, 2009).

2.5.1.2. İleri beslemeli yapay sinir ağları

İleri beslemeli yapay sinir ağları (Feedforward Neural Networks), en temel ve yaygın kullanılan sinir ağı mimarilerinden biridir. Bu ağlarda bilgi akışı yalnızca giriş katmanından çıkış katmanına doğru gerçekleşir. Ağ içerisinde geri besleme bağlantıları bulunmaz.

Her katmandaki nöronlar, yalnızca bir sonraki katmandaki nöronlara bağlanır. Bu yapı, matematiksel olarak modellenmesi ve eğitilmesi açısından oldukça basittir. İleri beslemeli yapay sinir ağları, özellikle sınıflandırma ve regresyon problemlerinde yaygın olarak kullanılmaktadır.

Bu mimari, homomorfik şifreleme gibi hesaplama kısıtlarının bulunduğu ortamlarda da tercih edilmekte olup, doğrusal ve doğrusal olmayan dönüşümlerin kontrollü bir şekilde uygulanmasına olanak tanımaktadır (Haykin, 2009).

2.5.1.3. Aktivasyon fonksiyonları

Aktivasyon fonksiyonları, yapay sinir ağlarının doğrusal olmayan problemleri öğrenebilmesini sağlayan temel bileşenlerdir. Bir nöronun toplam girdisini işleyerek çıktıya dönüştüren bu fonksiyonlar, ağın öğrenme kapasitesini doğrudan etkilemektedir.

Literatürde yaygın olarak kullanılan aktivasyon fonksiyonları arasında sigmoid, hiperbolik tanjant (tanh), ReLU ve doğrusal fonksiyonlar yer almaktadır. Bununla birlikte, yalnızca toplama ve çarpma işlemlerine izin veren homomorfik şifreleme tabanlı uygulamalarda, polinomsal aktivasyon fonksiyonları tercih edilmektedir.

Aktivasyon fonksiyonunun seçimi, ağın performansı, yakınsama hızı ve hesaplama maliyeti üzerinde belirleyici bir rol oynamaktadır (Haykin, 2009).

2.5.2. Yapay sinir ağlarında öğrenme

Yapay sinir ağlarında öğrenme, ağın ağırlık ve bias parametrelerinin hata fonksiyonunu minimize edecek şekilde güncellenmesi sürecidir. Bu süreç, genellikle denetimli öğrenme yaklaşımı ile gerçekleştirilir. Ağın ürettiği çıktı ile gerçek çıktı arasındaki fark hesaplanarak hata değeri elde edilir.

En yaygın öğrenme yöntemi geri yayılım algoritmasıdır (Backpropagation). Bu algoritmada hata değeri, zincir kuralı kullanılarak ağın çıkış katmanından giriş katmanına doğru yayılır ve ağırlıklar gradyan inişi yöntemiyle güncellenir.

Öğrenme süreci, ağın genelleme yeteneğini artırmayı hedeflerken aşırı öğrenmenin önlenmesi de büyük önem taşımaktadır. Bu nedenle öğrenme oranı, epoch sayısı ve ağ mimarisi dikkatli bir şekilde belirlenmelidir(Yılmaz ve Şimşek, 2026).

2.6. Homomorfik Şifreleme ve Makine Öğrenmesi

Şifrelenmiş verilerle yapılan makine öğrenimi, verilerin güvenliği ve gizliliğini korumak için önemli bir çözüm sunar. Ancak, bu tür veriler üzerinde yapılan analizler, birkaç teknik ve operasyonel zorlukla birlikte gelir. Şifrelemenin uygulanması, genellikle performans kayıpları, işlem zorlukları ve sınırlı araç kullanımı gibi sorunlara yol açabilir.

Kriptografi bilimin temel amaçlarından biri, şifrelenmiş verilerde istatistiksel saldırılardan korunmak için desen oluşumuna engel olmaktır. Düz metinlerde veya şifrelenmiş anahtardaki tek bir öğe değiştiğinde, şifrelenmiş metinde büyük değişiklere sebep olacağı beklenir. Bu durum şifreli metinlerde desen yakalama ihtimalini azaltır. Makine öğrenimi algoritmalarının şifrelenmiş veri setleri ile eğitilmeyeceği düşünülüyordu çünkü makine öğrenimi algoritmaları desenleri bulmaya çalışırken, kriptografik algoritmalar ise desen oluşumundan kaçınmaya çalışır. Yapılan çalışmalar homomorfik şifreleme yardımı ile şifrelenmiş veri setleri ile eğitilen makine öğrenmesi algoritmalarının diğer veri setleri için doğru çıkarımlar yapabileceği tespit edilmiştir. Makine öğrenmesi algoritmalarının başarılı sonuçlar çıkarması için veri setlerinin ihtimaller dahilinde büyük veri setleri ile çalışması gerekmektedir. Yani veri setlerinin büyüklüğü ne kadar artarsa makine öğrenmesi algoritmalarının çıkacağı sonuçların doğruluk oranları da o kadar artar (Kâhya, 2022).

Şifreleme, verilerin güvenliğini sağlamak için güçlü bir yöntem olsa da bu işlem genellikle veri boyutlarını artırır ve hesaplama gereksinimlerini artırabilir. Basitçe ifade etmek gerekirse: şifreleme, düz metne gürültü ekler ve böylece onu gizler. Şifre çözme

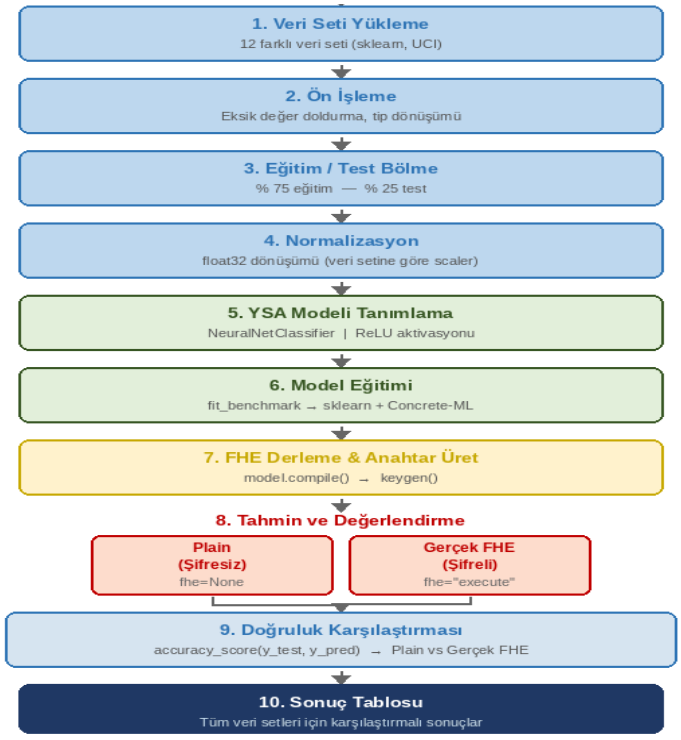
işlemi bu gürültüyü ortadan kaldırır. Şifrelenmiş veriler üzerinde gerçekleştirilen her işlem, şifreli metinlerin içindeki gürültüyü artırır. Gürültü belirli bir eşiği geçerse, doğru şifre çözme imkânsız hale gelir (Podschwadt vd., 2022). Şifrelenmiş veriler üzerinde yapılan işlemler, normal verilerle karşılaştırıldığında daha fazla işlem gücü gerektirir. Örneğin, homomorfik şifreleme gibi ileri düzey şifreleme teknikleri, şifrelenmiş veriler üzerinde doğrudan işlem yapılmasına olanak tanır, ancak bu işlemler, şifrelenmiş verilerin çözülmesinden çok daha yavaş ve hesaplama açısından pahalıdır. Bu durum, modelin eğitim sürecini önemli ölçüde yavaşlatabilir ve yüksek performanslı donanım gereksinimlerine yol açabilir.

Ayrıca, şifrelenmiş verilerle yapılan işlem süreçlerinde bellek kullanımı da artar. Verilerin şifreli biçimde tutulması, ek bellek alanı gerektirebilir ve bu da özellikle büyük veri setlerinde performans problemlerine yol açabilir. Şifrelenmiş veriler üzerinde makine öğrenimi uygulamalarını gerçekleştirmek için mevcut araçlar ve yöntemler henüz sınırlıdır. Çoğu makine öğrenimi algoritması, verilerin doğrudan erişilebilir olduğu varsayımıyla tasarlanmıştır ve bu algoritmaların şifreli veriler üzerinde doğrudan çalışması mümkün değildir.

Homomorfik şifreleme, büyük şifreli metinlerin boyutlarında, gürültü yönetiminde, yüksek hesaplama maliyetlerinde ve sınırlı işlevsellikte dahil olmak üzere zorluklar ve sınırlamalar sunar (Hamza, 2023). Şifrelenmiş veri işleme alanında sınırlı sayıda araç ve yöntem bulunmasının temel nedenleri, teknik zorluklar, performans kısıtlamaları, standartlaşma eksikliği ve güvenlik endişeleridir. Bu alandaki araçların ve yöntemlerin geliştirilmesi, veri gizliliği ve güvenliği konularındaki ilerlemelerle birlikte daha yaygın hale gelebilir. Araştırmacıların ve geliştiricilerin bu alanda yeni yöntemler ve optimizasyonlar üzerinde çalışmaları, şifrelenmiş veri işlemenin daha verimli ve uygulanabilir hale gelmesine katkıda bulunacaktır.

3. MATERYAL VE YÖNTEM

Bu çalışmada, verilerin gizliliğini bozmadan sınıflandırma işlemlerini gerçekleştirebilmek amacıyla Yapay Sinir Ağları (YSA) mimarisi ile Tam Homomorfik Şifreleme (FHE) teknikleri entegre edilmiştir. Uygulama aşamasında, Tamamen homomorfik şifreleme yöntemlerinden matematiksel olarak Torus yapısını temel alan TFHE şeması kullanılmış; bu şemanın makine öğrenmesi modellerine uygulanması sürecinde ise Concrete ML kütüphanesinden yararlanılmıştır.



Şekil 3.1. YSA+FHE deneysel çalışma akış diyagramı

Şekil 3.1'de sunulan akış diyagramı, uçtan uca deneysel sürecin genel çerçevesini ortaya koymaktadır. Süreç öncelikle veri setinin yüklenmesi ve ön işlenmesiyle başlamakta, eğitim ve test ayrımı ve normalizasyon adımlarının ardından Concrete-ML kütüphanesi aracılığıyla NeuralNetClassifier modelinin tanımlanması ve eğitilmesiyle devam etmektedir. Modelin eğitiminin tamamlanmasının ardından FHE derleme aşamasına geçilmekte ve bu aşamada şifreleme devresi oluşturulmakta ve kriptografik anahtarlar üretilmektedir. Son olarak şifresiz (plain) ve gerçek FHE çıkarımı olmak üzere iki farklı senaryo karşılaştırmalı olarak değerlendirilmekte; elde edilen doğruluk sonuçları tablo halinde sunulmaktadır. Bölümün devamındaki alt başlıklar bu akışın her

adımını ayrıntılı biçimde ele almaktadır. Bölümün devamındaki alt başlıklar bu akışın her adımını ayrıntılı biçimde ele almaktadır.

3.1. Kullanılan Veri Setleri

Bu çalışmada, homomorfik şifreleme altında yapay sinir ağı modellerinin şifreli çıkarım performansını kapsamlı bir biçimde değerlendirebilmek amacıyla, farklı alanlara ait toplam on adet gerçek dünya veri seti kullanılmıştır. Seçilen veri setleri; örnek sayısı, öznitelik boyutu ve problem türü açısından çeşitlilik göstermekte olup, homomorfik şifrelemenin farklı veri yapıları üzerindeki etkisini gözlemleyebilmek amacıyla tercih edilmiştir.

Kullanılan veri setleri, makine öğrenmesi alanında yaygın olarak kullanılan ve akademik çalışmalarda sıkça tercih edilen UCI Machine Learning Repository üzerinden temin edilmiştir. UCI veri setleri, standartlaştırılmış yapıları, açık erişilebilir olmaları ve farklı problem türlerini içermeleri nedeniyle karşılaştırmalı performans analizleri için uygun bir kaynak sunmaktadır. Seçilen veri setleri, farklı örnek sayıları ve özellik boyutlarına sahip olacak şekilde belirlenmiş olup küçük ve orta ölçekli veri yapılarında şifresiz ve tamamen homomorfik şifreleme (FHE) tabanlı yapay sinir ağı modellerinin performansının değerlendirilmesine olanak sağlamaktadır. Bu yaklaşım sayesinde, veri boyutunun ve özellik sayısının FHE tabanlı modeller üzerindeki etkisi ayrıntılı biçimde incelenmiştir. Veri setleri sınıflandırma problemlerine yönelik olup, her bir veri seti için giriş özellikleri ve hedef sınıf etiketleri açık bir şekilde tanımlanmıştır. Modelleme sürecinden önce veri setleri üzerinde gerekli ön işleme adımları uygulanmış; eksik veri kontrolü, ölçekleme ve normalizasyon işlemleri gerçekleştirilmiştir. Böylece hem şifresiz hem de şifreli modeller için adil ve tutarlı bir karşılaştırma ortamı oluşturulmuştur.

Parkinson veri seti, UCI Machine Learning Repository üzerinden elde edilen ve Parkinson hastalığının tespitine yönelik biyomedikal ses ölçümlerini içeren bir veri setidir. Veri seti, Parkinson hastalığına sahip bireyler ile sağlıklı bireylerin ayırt edilmesini amaçlayan bir sınıflandırma problemi olarak tanımlanmaktadır. Veri seti toplam 197 örnekten oluşmakta olup, her bir örnek bireylerden alınan bir ses kaydını temsil etmektedir. Veri setinde 22 adet sürekli öznitelik bulunmaktadır. Bu öznitelikler, temel frekans (Fo), frekans varyasyonu (jitter), genlik varyasyonu (shimmer), gürültü oranları (NHR, HNR) ve çeşitli doğrusal olmayan ölçümler gibi ses sinyaline ait biyomedikal özellikleri içermektedir. Veri setinde yer alan “status” değişkeni hedef

değişken olup, bireyin sağlık durumunu göstermektedir. Bu değişken, sağlıklı bireyler için 0, Parkinson hastalığına sahip bireyler için 1 değerini almaktadır. Veri setinde eksik değer bulunmaması, veri ön işleme sürecini kolaylaştırmaktadır. Parkinson veri seti, sağlık alanına ait hassas veriler içermesi nedeniyle veri gizliliği açısından kritik bir öneme sahiptir. Bu tür biyomedikal verilerin açık şekilde işlenmesi, hasta mahremiyetinin ihlal edilmesine yol açabileceğinden güvenli veri işleme yöntemlerine ihtiyaç duyulmaktadır (Dua ve Graff, 2019h).

Pima Indians Diabetes veri seti, UCI Machine Learning Repository üzerinden elde edilen ve bireylerin çeşitli tıbbi ölçümlerine dayanarak diyabet hastalığının tahmin edilmesini amaçlayan bir sınıflandırma veri setidir. Veri seti toplam 768 örnekten ve 8 adet öznitelikten oluşmaktadır. Tüm bireyler 21 yaş üstü Pima Kızılderili kökenli kadınlardan seçilmiştir. Veri setinde yer alan öznitelikler; gebelik sayısı, glikoz seviyesi, kan basıncı, cilt kalınlığı, insülin seviyesi, vücut kitle indeksi (BMI), diyabet soy ağacı fonksiyonu ve yaş gibi tıbbi ölçümleri içermektedir. Hedef değişken olan "Outcome", bireyin diyabet hastası olup olmadığını göstermektedir. Bu değişken 0 (sağlıklı) ve 1 (diyabet) olarak ikili sınıflandırma problemine karşılık gelmektedir. Veri setinde bazı özniteliklerde (örneğin glikoz, BMI vb.) yer alan sıfır değerler gerçekçi olmadığından eksik veri olarak değerlendirilmiş ve ortalama değerler ile doldurulmuştur. Bu veri seti, bireylere ait hassas sağlık bilgileri içerdiğinden veri gizliliği açısından kritik bir öneme sahiptir. Özellikle diyabet gibi kronik hastalıkların tahmini, kişisel sağlık verilerinin korunmasını zorunlu kılmaktadır. Bu çalışmada, homomorfik şifreleme yöntemi kullanılarak veriler şifreli halde işlenmiş ve böylece hasta verileri açığa çıkarılmadan makine öğrenmesi modeli uygulanmıştır. Bu yaklaşım, sağlık verilerinin güvenli şekilde analiz edilmesine olanak sağlamaktadır (Dua ve Graff, 2019i).

SAHeart veri seti, 462 örnek ve 9 öznitelikten oluşan, Güney Afrika kalp hastalığı verilerini içeren bir veri setidir. SAHeart (South African Heart Disease) veri seti, UCI Machine Learning Repository kapsamında yer alan ve bireylerde koroner kalp hastalığı (CHD) riskinin tahmin edilmesine yönelik oluşturulmuş bir veri setidir. Veri seti, Güney Afrika'daki bireylerden elde edilen klinik ve demografik özellikleri içermektedir. Toplamda 462 örnekten ve 9 öznitelikten oluşmaktadır. Bu öznitelikler arasında yaş, sigara kullanımı, LDL kolesterol seviyesi, obezite, alkol tüketimi ve ailede kalp hastalığı geçmişi gibi kalp hastalığı ile ilişkili önemli risk faktörleri bulunmaktadır. Veri setinde yer alan "chd" değişkeni hedef değişken olup, bireyin koroner kalp hastalığına sahip olup olmadığını göstermektedir. Bu değişken, 0 (hasta değil) ve 1 (hasta) şeklinde ikili

sınıflandırma problemi olarak ele alınmaktadır. Ayrıca veri setinde yer alan "famhist" (aile öyküsü) değişkeni kategorik olup, modelleme sürecinde sayısal forma dönüştürülerek kullanılmıştır. SAHeart veri seti, bireylerin sağlık durumları ve yaşam tarzı bilgilerini içermesi nedeniyle hassas veri kategorisinde değerlendirilmektedir. Özellikle kalp hastalığı gibi ciddi sağlık durumlarının tahmini, kişisel verilerin korunmasını zorunlu kılmaktadır. Çalışmada, homomorfik şifreleme yöntemi kullanılarak şifreli biçimde işlenmesi ve böylece bireylerin sağlık bilgileri açığa çıkarılmadan makine öğrenmesi modeli uygulanması için uygun bir veri seti olduğu için tercih edilmiştir (Hastie vd., 2009).

BUPA (Liver Disorders) veri seti, UCI Machine Learning Repository kapsamında yer alan ve bireylerde karaciğer hastalığı riskinin tahmin edilmesine yönelik oluşturulmuş bir veri setidir. Veri seti toplam 345 örnekten ve 6 adet öznitelikten oluşmaktadır. Bu öznitelikler, bireylerin kan testlerinden elde edilen biyokimyasal ölçümlerini içermekte olup karaciğer bozukluklarının biyokimyasal test sonuçlarına göre sınıflandırılmasını amaçlamaktadır. Özellikle karaciğer fonksiyonları ile ilişkili olan enzim değerleri (SGPT, SGOT, GGT vb.) ve alkol tüketimi (drinks) gibi değişkenler veri setinde yer almaktadır. Veri setinde yer alan "selector" değişkeni hedef değişken olup, bireyin karaciğer hastalığına sahip olup olmadığını dolaylı olarak temsil etmektedir. Bu değişken 1 ve 2 değerlerinden oluştuğu için ikili sınıflandırma problemine uygun şekilde 0 ve 1 değerlerine dönüştürülmüştür. BUPA veri seti, bireylerin sağlık durumlarına ilişkin biyokimyasal veriler içermesi nedeniyle hassas veri kategorisinde değerlendirilmektedir. Bu tür verilerin güvenli şekilde işlenmesi, özellikle sağlık alanında veri gizliliğinin korunması açısından önem taşıdığı için şifrelemiş verilerle çalışmak için uygun görülmüştür(Dua ve Graff, 2019b).

Heart Disease veri seti, UCI Machine Learning Repository bünyesinde yer alan ve bireylerde kalp hastalığının varlığını tahmin edilmesini amaçlayan bir sağlık veri setidir. Veri seti; Cleveland, Macaristan, İsviçre ve VA Long Beach olmak üzere dört farklı kaynaktan elde edilen verileri içermektedir. Ancak literatürde en yaygın kullanılan ve bu çalışmada tercih edilen veri seti Cleveland veri setidir. Veri seti toplam 303 örnekten ve 13 öznitelikten oluşmaktadır. Bu öznitelikler; yaş, cinsiyet, göğüs ağrısı tipi, kan basıncı, kolesterol seviyesi, maksimum kalp atış hızı gibi kalp hastalığı ile ilişkili klinik ve demografik bilgileri içermektedir. Hedef değişken olan "num", hastada kalp hastalığının varlığını ifade etmektedir. Bu değişken 0 ile 4 arasında değerler almakta olup, bu çalışmada literatüre uygun olarak 0 (hastalık yok) ve 1 (hastalık var) şeklinde ikili

sınıflandırma problemine dönüştürülmüştür. Veri setinde eksik değerler bulunduğundan, bu değerler ortalama ile doldurularak modelleme sürecine dahil edilmiştir. Heart Disease veri seti, bireylerin sağlık durumlarına ilişkin kritik bilgiler içermesi nedeniyle veri gizliliği açısından yüksek öneme sahiptir. Kalp hastalığı gibi ciddi sağlık verilerinin açık şekilde işlenmesi, birey mahremiyetini riske atabilmektedir. Çalışmada, homomorfik şifreleme yöntemi kullanılarak veriler şifreli biçimde işlenmesine ve böylece hassas sağlık verileri korunarak makine öğrenmesi modeli uygulanmasına uygun olduğu için kullanılmıştır (Dua ve Graff, 2019f).

Credit Approval veri seti, UCI Machine Learning Repository kapsamında yer alan ve bireylerin kredi başvurularının onaylanıp onaylanmayacağını tahmin etmeye yönelik oluşturulmuş bir sınıflandırma veri setidir. Veri seti toplam 690 örnekten ve 15 öznitelikten oluşmaktadır. Bu öznitelikler, başvuru sahibine ait demografik ve finansal bilgileri içermektedir. Veri setinde hem sayısal hem de kategorik değişkenler birlikte yer almakta olup, bu durum veri ön işleme sürecini daha önemli hale getirmektedir. Hedef değişken olan sınıf etiketi, başvurunun onaylanması (+) veya reddedilmesi (-) durumunu ifade etmektedir. Bu çalışmada, hedef değişken ikili sınıflandırma problemine uygun olarak 0 ve 1 değerlerine dönüştürülmüştür. Credit Approval veri setinde yer alan kategorik değişkenler one-hot encoding yöntemiyle sayısal forma dönüştürülmüş, eksik değerler uygun istatistiksel yöntemlerle tamamlanmış ve veriler standartlaştırılarak modele uygun hale getirilmiştir. Credit Approval veri seti, bireylerin finansal durumlarına ilişkin hassas bilgiler içermektedir. Bu tür verilerin açık şekilde işlenmesi, kişisel veri güvenliği açısından risk oluşturabilmektedir. Bu çalışmada, homomorfik şifreleme yöntemi kullanılarak veriler şifreli biçimde işlenmesi ve böylece bireylerin finansal bilgileri korunarak makine öğrenmesi modeli uygulanması için uygundur. Bu yaklaşım, özellikle finans sektöründe güvenli veri analizi açısından önemli bir avantaj sağlamaktadır(Dua ve Graff, 2019c) .

German Credit (Statlog) veri seti, bireylerin finansal özelliklerine dayanarak kredi risklerinin sınıflandırılmasını amaçlamaktadır. Veri seti, kapsamında sunulmuş olup sosyal bilimler alanında yaygın olarak kullanılan çok değişkenli bir veri setidir. Veri seti toplam 1000 örnek ve 20 özellikten oluşmaktadır. Özellikler; bireylerin kredi geçmişi, mevcut hesap durumu, kredi miktarı, çalışma süresi, yaş, konut durumu ve benzeri finansal ve demografik bilgileri içermektedir. Veri setinde hem kategorik hem de sayısal değişkenler bulunmaktadır ve herhangi bir eksik veri içermemektedir. Veri setinin hedef değişkeni, bireylerin kredi risk durumunu ifade etmekte olup iki sınıftan oluşmaktadır:

iyi kredi (good) ve kötü kredi (bad). Bu sınıflandırma problemi, makine öğrenmesi literatüründe yaygın olarak kullanılan ikili sınıflandırma problemlerinden biridir. Ayrıca veri seti, yanlış sınıflandırmaların maliyetlerinin farklı olduğu bir maliyet matrisi ile birlikte sunulmaktadır. Veri setinin hem kategorik hem de sayısal değişkenler içermesi nedeniyle, modelleme sürecinde uygun ön işleme adımları uygulanmıştır. Kategorik değişkenler sayısal forma dönüştürülmüş, veriler ölçeklendirilmiş ve makine öğrenmesi modelleri için uygun hale getirilmiştir. German Credit veri seti, kredi risk analizi ve finansal karar destek sistemleri alanında model performansını değerlendirmek amacıyla literatürde sıklıkla tercih edilen bir benchmark veri setidir. Bu sebeple şifreleme gereken bir alan olması ve yapay sinir ağları için uygun bir model olduğu için tercih edilmektedir(Dua ve Graff, 2019d).

Wisconsin Diagnostic Breast Cancer (WDBC) veri seti, meme kanserinin iyi huylu (benign) ve kötü huylu (malignant) olarak sınıflandırılmasını amaçlayan yaygın bir veri setidir. Veri seti, Wisconsin Üniversitesi tarafından sağlanmış olup makine öğrenmesi çalışmalarında sıklıkla kullanılmaktadır. Veri seti toplam 569 örnekten ve her biri hücre çekirdeğine ait özellikleri temsil eden 30 sayısal öznitelikten oluşmaktadır. Bu öznitelikler; hücrelerin yarıçapı, dokusu, çevresi, alanı, pürüzsüzlüğü, kompaktlığı ve simetrisi gibi morfolojik özellikleri içermektedir. Hedef değişken ikili yapıdadır ve örneklerin iyi huylu (0) veya kötü huylu (1) olduğunu ifade etmektedir. Veri setinde eksik değer bulunmamaktadır. Bu veri setinde tüm özellikler sayısal olduğu için ek bir kategorik dönüşüm işlemine ihtiyaç duyulmamıştır. Veriler modele uygun hale getirilmeden önce standardizasyon işlemine tabi tutulmuş ve ardından eğitim ve test kümelerine ayrılmıştır. Bu veri seti de sağlık verileri içerdiği için şifrelenmeye ve yapay sinir ağları yapmak için uygun bulunmuştur (Dua ve Graff, 2019j).

Breast Cancer veri seti, Yugoslavya Ljubljana Üniversitesi Tıp Merkezi Onkoloji Enstitüsü tarafından sağlanmış olup meme kanseri hastalarında hastalığın tekrarlama durumunu tahmin etmeye yönelik bir sınıflandırma problemidir. Bu veri seti, makine öğrenmesi literatüründe yaygın olarak kullanılan klasik veri setlerinden biridir. Veri seti toplam 286 örnekten oluşmakta olup her örnek, hastaya ait klinik ve demografik bilgileri temsil eden 9 öznitelik ile tanımlanmaktadır. Hedef değişken ikili yapıdadır ve hastalığın tekrarlamayan (no-recurrence-events) veya tekrarlayan (recurrence-events) olduğunu ifade etmektedir. Veri setinde bazı özniteliklerde eksik değerler bulunmaktadır. Özelliklerin büyük çoğunluğu kategorik yapıda olduğundan, modelleme öncesinde kategorik değişkenler one-hot encoding yöntemi ile sayısal forma dönüştürülmüştür.

Eksik veriler uygun istatistiksel yöntemlerle (en sık gözlenen değer ile) tamamlanmış ve tüm veriler standardizasyon işleminden geçirilerek modele uygun hale getirilmiştir (Dua ve Graff, 2019a).

Haberman's Survival veri seti, meme kanseri nedeniyle ameliyat geçiren hastaların uzun dönem hayatta kalma durumlarını incelemek amacıyla oluşturulmuş klasik bir sınıflandırma veri setidir. Veri seti, 1958–1970 yılları arasında Chicago Üniversitesi Billings Hastanesi'nde tedavi gören hastalara ait klinik verilerden elde edilmiştir. Veri seti toplam 306 örnekten ve her biri hastaya ait klinik bilgileri temsil eden 3 sayısal öznitelikten oluşmaktadır. Bu öznitelikler; hastanın ameliyat sırasındaki yaşı, ameliyat yılı ve pozitif aksiller lenf düğümü sayısını içermektedir. Hedef değişken ikili yapıdadır ve hastanın 5 yıl veya daha uzun süre hayatta kalıp kalmadığını ifade etmektedir. Veri setinde eksik değer bulunmamaktadır. Tüm öznitelikler sayısal olduğu için veri üzerinde ek bir kategorik dönüşüm uygulanmamış, yalnızca model performansını artırmak amacıyla standardizasyon işlemi gerçekleştirilmiştir. Daha sonra veri eğitim ve test kümelerine ayrılarak modelleme sürecine dahil edilmiştir (Dua ve Graff, 2019e).

İris veri seti, Ronald A. Fisher tarafından 1936 yılında önerilen ve makine öğrenmesi literatüründe en yaygın kullanılan veri setlerinden biridir. Veri seti, sınıflandırma problemlerinde algoritmaların performansını değerlendirmek amacıyla sıklıkla tercih edilmektedir. Veri seti toplam 150 örnekten oluşmakta olup, her bir örnek bir iris bitkisinin çiçek türlerinin sınıflandırılmasına yönelik ve makine öğrenmesi literatüründe sıklıkla referans verilen temel veri setlerinden biridir. Örnekler üç farklı sınıfa ayrılmaktadır. Bunlar iris Setosa, iris versicolour ve iris virginica. Her bir sınıf 50 örnek içermektedir. Veri setinde dört adet sürekli öznitelik bulunmaktadır. Bu öznitelikler; çanak yaprağı uzunluğu, çanak yaprağı genişliği, taç yaprağı uzunluğu ve taç yaprağı genişliği olup santimetre cinsinden ölçülmektedir. Veri setinde eksik değer bulunmaması ve düşük boyutlu yapısı, modelleme sürecini kolaylaştırmaktadır. İris veri setinin önemli bir özelliği, sınıflar arasındaki ayrımın kısmen doğrusal olmasıdır. Setosa sınıfı diğer sınıflardan kolaylıkla ayrılabilirken, versicolour ve virginica sınıfları arasında ayrım daha zordur. Bu durum, veri setini hem basit hem de model performansını değerlendirmek açısından anlamlı bir problem haline getirmektedir. Bu çalışmada İris veri seti, homomorfik şifreleme yöntemlerinin makine öğrenmesi üzerindeki etkisini gözlemlemek amacıyla kullanılmıştır. Veri setinin düşük boyutlu ve dengeli yapısı,

şifrelenmiş veri üzerinde gerçekleştirilen işlemlerin doğruluğunu analiz etmek için uygun bir ortam sunmaktadır (Dua ve Graff, 2019g).

Wine veri seti, farklı üzüm türlerinden elde edilen şarapların kimyasal analiz sonuçlarına dayanarak sınıflandırılmasını amaçlayan, makine öğrenmesi literatüründe yaygın olarak kullanılan bir veri setidir. Veri seti, UCI Machine Learning Repository üzerinden temin edilmiştir. Veri seti toplam 178 örnekten oluşmakta olup, her bir örnek bir şarap numunesini temsil etmektedir. Örnekler üç farklı sınıfa ayrılmaktadır ve her sınıf farklı bir üzüm türünden elde edilen şarapları kimyasal analiz sonuçlarına dayanarak şarap türlerinin sınıflandırılmasını amaçlayan çok sınıflı bir veri setidir. Veri setinde toplam 13 adet sürekli öznitelik bulunmaktadır. Bu öznitelikler; alkol oranı, malik asit, kül, magnezyum, toplam fenoller, flavonoidler, renk yoğunluğu ve prolin gibi şarabın kimyasal özelliklerini temsil eden değişkenlerden oluşmaktadır. Veri setinde eksik değer bulunmamakta olup, tüm özellikler sayısal yapıda olduğundan veri ön işleme sürecini kolaylaştırmakta ve modelleme aşamasında doğrudan kullanılmaktadır. Ayrıca sınıflar arasındaki ayrımın belirgin olması, bu veri setini sınıflandırma algoritmalarının performansını değerlendirmek için uygun bir test ortamı haline getirmektedir. Wine veri seti, tamamen sayısal özelliklerden oluşması nedeniyle homomorfik şifreleme tabanlı makine öğrenmesi uygulamaları için oldukça uygun bir veri setidir. Özellikle çok boyutlu yapısı sayesinde, şifreli veriler üzerinde gerçekleştirilen hesaplamaların model performansına etkisini değerlendirmek mümkün olmaktadır (Dua ve Graff, 2019k).

Çizelge 3.1. Kullanılan veri setleri ve ayırıcı özellikleri

Veri Seti	Sınıf Sayısı	Özellik Sayısı	Örnek Sayısı	Veri Tipi
Parkinson	2	22	197	Sürekli
Pima Diabetes	2	8	768	Sürekli
SAHeart	2	9	462	Karma
BUPA	2	6	345	Sürekli
Heart Disease	2	13	303	Karma
Credit Approval	2	15	690	Karma
German Credit	2	20	1000	Karma
WDBC (Breast Cancer)	2	30	569	Sürekli
Breast Cancer (Categorical)	2	9	286	Kategorik
Habermans's Survival	2	3	306	Tamsayı
İris	3	4	150	Sürekli
Wine	3	13	178	Sürekli

3.2. Veri Ön İşleme

Bu çalışmada kullanılan 12 veri seti model eğitimi ve performans değerlendirmesi öncesinde sistematik bir veri ön işleme sürecinden geçirilmiştir. Bu ön işleme sürecinde veri temizleme yapılmış ardından kategorik verilerin dönüştürülmesi gerçekleştirilmiş sonrasında ise eksik değerlerin giderilmesi sağlanmış ardından veri bölme ve normalizasyon işlemleri gerçekleştirilmiştir. Bu işlemler yapay sinir ağı modellerinin sağlıklı bir şekilde eğitilebilmesi ve homomorfik şifreleme altında gerçekleştirilen çıkarım sürecinin doğru biçimde değerlendirilebilmesi amacıyla veri ön işleme adımları uygulanmıştır.

İlk aşamada, veri setleri ilgili kaynaklardan yüklenmiş genellikle uci ve kaggle tercih edilmiş sonrasında veriler analiz edilmiştir. Veri setlerinde yer alan ve modele katkı sağlamayan indeks niteliğindeki sütunlar veri setinden çıkarılmıştır. Kategorik özellikler, makine öğrenmesi algoritmaları tarafından işlenebilir hale getirilmek amacıyla sayısal formata dönüştürülmüştür. Örneğin, SAHeart veri setinde yer alan famhist değişkeni “Present” ve “Absent” değerlerine karşılık gelecek şekilde ikilik taban (binary) biçimde kodlanmıştır.

İlk aşamada, veri setlerinde yer alan eksik veya tutarsız gözlemler kontrol edilmiş ve ilgili veri setlerinin yapısına uygun şekilde ele alınmıştır. Eksik değer içeren öznitelikler, veri setinin genel dağılımını bozmayacak biçimde ortalama veya medyan değerler kullanılarak doldurulmuştur. Kategorik özellik içeren veri setlerinde, bu özellikler sayısal forma dönüştürülerek yapay sinir ağı modelleri için uygun hale getirilmiştir. Sonrasında kullanılan veri setlerinde eksik değerlerin bulunması durumunda, bu değerler model performansını olumsuz etkilememesi için ele alınmıştır. Eksik veriler, ilgili özelliğin ortalama değeri ile doldurulmuştur. Bu yaklaşım, veri kaybını önleyerek modelin daha kararlı ve tutarlı sonuçlar üretmesine katkı sağlamaktadır.

Makine öğrenmesi modellerinde performans değerlendirmesi amacıyla veri setleri eğitim ve test kümelerine ayrılmaktadır. Modelin oluşturulmasında kullanılan veri seti ile performans değerlendirilmesi yapılamaz çünkü model, eğitim yapılan verileri ezberleyebilir ve eğitim kümesi üzerindeki örnekler için doğru tahminler üretebilir ancak bu durum modelin ezberlemesinden kaynaklanır. Daha önce görmediği veriler ile performans değerlendirilmesi yapılmak istendiğindeki başarısını gösterememektedir. Bu nedenle veri seti iki bölüme ayrılmaktadır. İlk bölüm modelin öğrenme sürecinde

kullanılan eğitim kümesi (training set), ikinci bölüm ise modelin daha önce görmediği veriler üzerindeki performansını değerlendirmek amacıyla kullanılan test kümesi (test set) olarak ayrılmaktadır. Veri setinin eğitim ve test kümelerine ayrılması işleminde train_test_split fonksiyonu kullanılmaktadır. Bu işlem sırasında veri setinin %75'i eğitim kümesi, %25'i ise test kümesi olarak ayrılmıştır. Eğitim ve test verilerinin oranı kesin bir kurala bağlı olmamakla birlikte, test verisinin toplam verinin %25'ini oluşturması yaygın olarak kullanılan bir yaklaşımdır. Veri seti eğitim ve test olarak ayrılmadan önce karıştırılmaktadır. Bunun nedeni veri setinde bulunan örneklerin belirli sınıflara göre sıralı olabilmesidir. Veri karşılaştırma işlemi sayesinde eğitim ve test kümelerinde farklı sınıflara ait örneklerin bulunması sağlanmaktadır. Ayrıca yapılan deneylerin tekrarlanabilirliğini sağlamak amacıyla random_state parametresi yani rastgelelik parametresi kullanılarak rastgele sayı üretecine sabit bir başlangıç değeri verilmiştir. Yapay sinir ağı modelleri, şifresiz ortamda yalnızca eğitim verileri kullanılarak eğitilmiş, test verileri ise hem şifresiz hem de homomorfik şifreleme altında şifrelenmiş olarak çıkarım sürecinde kullanılmıştır. Bu yaklaşım sayesinde, aynı test verileri üzerinde şifreli ve şifresiz senaryoların performansları doğrudan karşılaştırılabilmiştir (Müller ve Guido, 2016).

Veri ön işlemenin bir diğer önemli adımı, özniteliklerin ölçeklendirilmesidir. Yapay sinir ağları ve homomorfik şifreleme tabanlı hesaplamalar, özniteliklerin benzer ölçeklerde olmasını gerektirmektedir. Veri ön işleme kısmında, özelliklerin farklı ölçeklerde olmasının model performansı üzerindeki olumsuz etkilerini ortadan kaldırmak amacıyla normalizasyon işlemi uygulanmıştır. Veri setlerinde yer alan özniteliklerin farklı değer aralıklarına sahip olması, makine öğrenmesi modellerinin performansını etkileyebilmektedir. Özellikle geniş değer aralıklarına sahip öznitelikler model üzerinde daha baskın hale gelebilmekte ve bu durum öğrenme sürecini olumsuz etkileyebilmektedir. Bu nedenle veri ön işleme aşamasında normalizasyon işlemleri uygulanmaktadır. Normalizasyon işlemi, verilerin belirli bir aralığa dönüştürülerek özniteliklerin ortak ölçekte temsil edilmesini sağlamaktadır. Bu yöntem özellikle yapay sinir ağları ve uzaklık tabanlı makine öğrenmesi algoritmalarında yaygın olarak kullanılmaktadır (Han vd., 2011). Bu çalışmada veri ön işleme aşamasında Min-Max normalizasyonu ve Z-Score standartlaştırma yöntemleri kullanılmıştır.

Min-Max normalizasyonu, veriler üzerinde doğrusal bir dönüşüm gerçekleştirerek öznitelik değerlerini belirli bir aralığa dönüştürmektedir. Bu yöntemde veri değerleri genellikle [0,1] veya [-1,1] aralığında yeniden ölçeklendirilmektedir(Han et al., 2011).

Bahsi geçen Min-Max normalizasyon yönteminin matematiksel formülü Denklem (3.1)'de verilmiştir.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3.1}$$

Burada x mevcut veri değerini, x_{min} ilgili özniteliğin minimum değerini, x_{max} ise maksimum değerini ifade etmektedir. x' ise normalizasyon işlemi sonucunda elde edilen yeni veri değeridir. Bu yöntem veri değerleri arasındaki ilişkileri koruyarak yeniden ölçeklendirme işlemi gerçekleştirmektedir(Han vd., 2011).

Z-Score standardizasyonunda ise veri değerleri ilgili özniteliğin ortalaması ve standart sapması kullanılarak dönüştürülmektedir. Bu yöntem sonucunda verilerin ortalaması 0, standart sapması ise 1 olmaktadır (Han vd., 2011). Z-Score standardizasyonunun matematiksel gösterimi Denklem (3.2)'de verilmiştir.

$$v_i' = \frac{v_i - \bar{A}}{\sigma_A} \tag{3.2}$$

Z-Score standardizasyonunda $\bar{A}$ ilgili özniteliğin ortalamasını, σ_A ise standart sapmasını ifade etmektedir. Z-Score normalizasyonu özellikle özniteliğin minimum ve maksimum değerlerinin bilinmediği veya uç değerlerin Min-Max normalizasyonunu baskıladığı durumlarda faydalı olmaktadır (Han vd., 2011). Bu çalışmada temel olarak Z-score standardizasyonu kullanılmıştır. Standardizasyon işlemi yalnızca eğitim verisi üzerinde öğrenilmiş ve elde edilen dönüşüm parametreleri test verisine uygulanmıştır. Bu yaklaşım, veri sızıntısını önlemek açısından kritik öneme sahiptir. Bununla birlikte, farklı normalizasyon yöntemlerinin model performansı üzerindeki etkisini incelemek amacıyla Min-Max ölçekleme yöntemleri de kullanılmıştır. Bu kapsamda veriler genellikle [0,1] veya [-1,1] aralıklarına dönüştürülerek farklı ölçekleme teknikleri altında karşılaştırmalı analizler gerçekleştirilmiştir.

Homomorfik şifreleme tabanlı modellerde kullanılmak üzere, veriler ayrıca nicemleme (quantization) sürecine uygun hale getirilmiştir. Bu süreçte sürekli değerler belirli bir bit genişliği ile temsil edilebilir forma dönüştürülmüş ve şifreli hesaplama işlemleri için uygun veri yapısı oluşturulmuştur. Nicemleme (quantization)işlemi hesaplama maliyetini ve model doğruluğunu doğrudan etkileyen önemli bir aşamadır. Model uyumluluğunu sağlamak ve hesaplama verimliliğini artırmak amacıyla veri tipleri

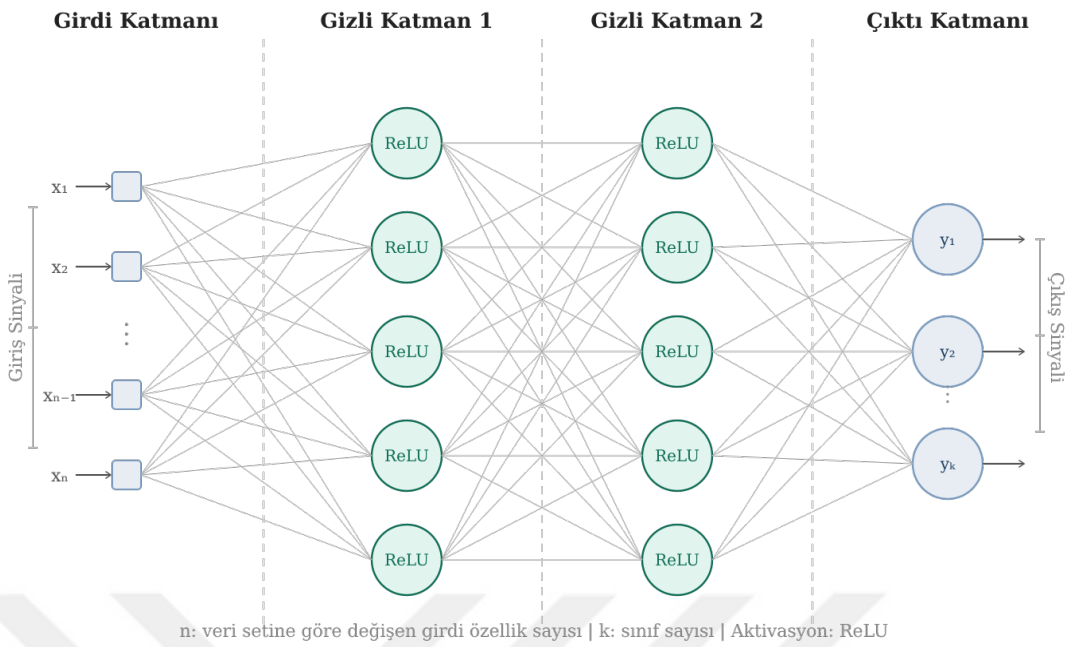
çeşitli dönüşümler yapılmıştır. Bu dönüşümler, özellikle Concrete-ML kütüphanesi ile gerçekleştirilen şifreli işlemlerde performans açısından avantaj sağlamaktadır.

Son olarak, tüm veri ön işleme adımları hem Scikit-Learn tabanlı şifresiz modeller hem de Concrete-ML kullanılarak gerçekleştirilen şifreli çıkarım senaryoları için tutarlı biçimde uygulanmış, böylece elde edilen sonuçların adil ve karşılaştırılabilir olması sağlanmıştır.

3.3. Yapay Sinir Ağı Modeli (Scikit-learn)

Bu çalışmada, homomorfik şifreleme altında gerçekleştirilen şifreli çıkarım sonuçlarını değerlendirebilmek amacıyla, referans model olarak şifresiz ortamda eğitilen bir yapay sinir ağı modeli kullanılmıştır. Şifresiz model, Scikit-Learn kütüphanesi içerisinde yer alan çok katmanlı algılayıcı (Multi-Layer Perceptron – MLP) yapısı kullanılarak oluşturulmuştur. Bu model, homomorfik şifreleme altında elde edilen sonuçların doğruluk ve performans açısından karşılaştırılabilmesi için temel bir karşılaştırma noktası olarak değerlendirilmiştir.

Kullanılan yapay sinir ağı modeli ileri beslemeli bir mimariye sahip olup, giriş katmanı, bir veya daha fazla gizli katman ve bir çıkış katmanından oluşmaktadır. Giriş katmanındaki nöron sayısı, ilgili veri setinin öznitelik sayısına bağlı olarak belirlenmiştir. Gizli katmanlarda yer alan nöron sayısı ve katman sayısı, deneysel çalışmalar sonucunda dengeli bir doğruluk ve hesaplama maliyeti sağlayacak şekilde seçilmiştir. Çıkış katmanında ise, sınıflandırma probleminin türüne bağlı olarak uygun sayıda nöron kullanılmıştır. Modelin eğitim sürecinde, ağırlıkların güncellenmesi geri yayılım (backpropagation) algoritması ile gerçekleştirilmiştir. Hata fonksiyonu olarak çapraz entropi kaybı kullanılmış ve optimizasyon işlemi, gradyan tabanlı bir yaklaşım ile yürütülmüştür. Eğitim sırasında, aşırı öğrenmenin önüne geçebilmek amacıyla belirli sayıda epoch ve uygun bir öğrenme oranı kullanılmıştır.



Şekil 3.2. Yapay sinir ağı mimarisi (Katman sayısı: 2 veya 3)

Şekil 3.1'de çalışmada kullanılan temsili yapay sinir ağı (YSA) mimarisi gösterilmektedir. Model yapısı; giriş katmanı, gizli katmanlar ve çıkış katmanından oluşan çok katmanlı ileri beslemeli yapay sinir ağı yapısına sahiptir. Giriş katmanındaki nöron sayısı (n), ilgili veri setindeki öznitelik sayısına göre belirlenmiştir. Veri setine ait özellikler giriş katmanına aktarılmakta ve ağırlıklı bağlantılar aracılığıyla gizli katmanlara iletilmektedir. Çalışmada kullanılan gizli katman yapıları iki veya üç katmandan oluşacak şekilde yapılandırılmıştır. Gizli katmanlardaki nöron sayıları, giriş özellik sayısının belirli bir çarpan ile genişletilmesi prensibine dayanmaktadır. Bu kapsamda $n \times 3$ve $n \times 5$çarpanları kullanılmıştır. Gizli katmanlarda doğrusal olmayan öğrenme yeteneği sağlayabilmek amacıyla ReLU (Rectified Linear Unit) aktivasyon fonksiyonu tercih edilmiştir. Çıkış katmanındaki nöron sayısı (k), veri setindeki sınıf sayısına göre belirlenmiştir. İkili sınıflandırma problemlerinde iki çıkış nöronu kullanılırken, çok sınıflı veri setlerinde sınıf sayısına uygun çıkış yapısı oluşturulmuştur. Model eğitim süreçlerinde Concrete-ML kütüphanesinde bulunan NeuralNetClassifier yapısı kullanılmış olup katman sayısı, gizli nöron çarpanı ve epoch değerleri veri setlerinin özelliklerine göre yapılandırılmıştır.

Bu çalışmada kullanılan yapay sinir ağı modelinin mimarisi, 12 farklı veri seti ve sentetik veriler üzerinde gerçekleştirilen kapsamlı deneysel analizler sonucunda belirlenmiştir. Model tasarımında özellikle katman sayısı ve gizli katmanlardaki nöron

sayısı parametrelerinin model performansı üzerindeki etkisi detaylı olarak incelenmiştir. Deneysel sonuçlar genel olarak değerlendirildiğinde, iki katmanlı yapıların tüm veri setlerinde yüksek ve dengeli performans sağladığı görülmüştür. Üç katmanlı modellerin bazı veri setlerinde benzer doğruluk değerlerine ulaştığı, ancak çoğu durumda performans üzerinde anlamlı bir iyileşme sağlamadığı tespit edilmiştir. Buna ek olarak, katman sayısının artırılması model karmaşıklığını ve hesaplama maliyetini artırmasına rağmen, doğruluk üzerinde kayda değer bir katkı sunmamıştır. Bu nedenle, model mimarisinde daha sade ve hesaplama açısından daha verimli olan iki katmanlı yapılar tercih edilmiştir. Gizli katmanlardaki nöron sayısı açısından yapılan değerlendirmelerde ise, veri setlerinin yapısına bağlı olarak farklı ihtiyaçlar ortaya çıktığı gözlemlenmiştir. Sınıflar arası ayrımın daha belirgin olduğu ve daha düşük boyutlu veri setlerinde (örneğin İris ve Parkinsons), 3 nöronlu yapıların yüksek doğruluk değerlerine ulaşmak için yeterli olduğu belirlenmiştir. Buna karşılık, daha karmaşık ve örüntü yapısı daha zor öğrenilen veri setlerinde (Heart Disease, SAHeart ve Bupa vs.), 5 nöronlu yapıların daha başarılı sonuçlar verdiği görülmüştür. Bu durum, model kapasitesinin veri setinin karmaşıklığına uygun şekilde ayarlanmasının önemini açıkça ortaya koymaktadır.

Ayrıca, bazı veri setlerinde nöron sayısının artırılmasının doğruluk üzerinde sınırlı bir etki yarattığı veya performansı olumsuz etkilediği gözlemlenmiştir. Bu bulgu, gereğinden fazla parametre kullanımının her zaman daha iyi sonuçlar üretmediğini ve aşırı model karmaşıklığının genelleme performansını olumsuz etkileyebileceğini göstermektedir.

Sonuç olarak, bu çalışmada model mimarisi belirlenirken yalnızca doğruluk performansı değil, aynı zamanda model karmaşıklığı ve hesaplama maliyeti de dikkate alınmıştır. Bu kapsamda, veri setine özgü olarak en uygun katman ve nöron sayıları belirlenmiş ve performans ile verimlilik arasında dengeli bir yapı oluşturulmuştur. Elde edilen bu şifresiz model sonuçları, homomorfik şifreleme altında gerçekleştirilen model performansının değerlendirilmesi için temel bir referans noktası oluşturmuştur.

3.3.1. Model mimarisi

Bu çalışmada kullanılan şifresiz yapay sinir ağı modeli, ileri beslemeli çok katmanlı algılayıcı (Multi-Layer Perceptron – MLP) mimarisi temel alınarak tasarlanmıştır. Seçilen mimari, homomorfik şifreleme altında gerçekleştirilen şifreli çıkarım süreci ile uyumlu olacak şekilde hem hesaplama karmaşıklığı hem de model

doğruluğu açısından dengeli bir yapı sunmaktadır. Model mimarisi oluşturulurken, gizli katmanlardaki nöron sayısı sabit bir değer olarak belirlenmemiş, bunun yerine veri setinin giriş boyutuna bağlı olarak değişen bir yapı tercih edilmiştir. Model, bir giriş katmanı, bir gizli katman ve bir çıkış katmanından oluşmaktadır. Giriş katmanındaki nöron sayısı, her bir veri setinin sahip olduğu öznitelik sayısına bağlı olarak belirlenmiştir. Bu yaklaşım, farklı boyutlardaki veri setleri için modelin esnek bir şekilde kullanılabilmesini sağlamaktadır.

Bu parametreye göre, gizli katmanlardaki nöron sayısı veri setinin özellik (feature) sayısı ile belirlenen bir katsayının çarpımı şeklinde hesaplanmaktadır. Örneğin, 10 özellikten oluşan bir veri seti için ve çarpan değeri 5 olarak seçildiğinde, gizli katmandaki nöron sayısı 50 olmaktadır. Benzer şekilde, 5 özellik içeren bir veri setinde ise bu değer 25 olarak belirlenmektedir. Bu yaklaşım sayesinde model mimarisi, farklı boyutlardaki veri setlerine uyum sağlayabilecek esnek bir yapıya kavuşmaktadır. Sabit nöron sayısı kullanılması durumun da bazı veri setleri için yetersiz öğrenme (underfitting), bazıları için ise gereksiz model karmaşıklığı ortaya çıkabilmektedir. Buna karşılık, giriş boyutuna bağlı olarak ölçeklenen bu yapı, model kapasitesinin veri setinin büyüklüğü ile orantılı olarak ayarlanmasına olanak tanımaktadır. Bunun sonucunda nöron sayısının veri setine göre dinamik olarak belirlenmesi model performansını artırmakta ve gereksiz hesaplama maliyetinin önüne geçmektedir.

Gizli katmanda doğrusal olmayan ilişkilerin öğrenilebilmesi amacıyla ReLU (Rectified Linear Unit) aktivasyon fonksiyonu kullanılmıştır. ReLU fonksiyonu, hesaplama verimliliği ve yaygın kullanımı nedeniyle tercih edilmiştir. Çıkış katmanında ise, sınıflandırma probleminin türüne bağlı olarak uygun bir aktivasyon fonksiyonu kullanılmıştır. İkili sınıflandırma problemleri için sigmoid fonksiyonu, çok sınıflı problemler için ise softmax fonksiyonu tercih edilmiştir.

Seçilen model mimarisi, Concrete-ML kütüphanesinin desteklediği yapılarla uyumlu olacak şekilde belirlenmiştir. Özellikle, homomorfik şifreleme altında doğrusal olmayan fonksiyonların doğrudan hesaplanamaması nedeniyle, modelin şifreli çıkarım aşamasında kullanılabilirliği göz önünde bulundurulmuştur. Bu doğrultuda, aktivasyon fonksiyonları Concrete-ML tarafından polinom yaklaşımlarla temsil edilebilecek biçimde ele alınmıştır.

Çizelge 3.2. Veri setlerine göre yapay sinir ağı model parametreleri

Veri seti	Girdi (n)	Katman sayısı	Nöron çarpanı	Gizli nöron sayısı	Epoch	Çıktı (k)
Wine	13	2	3	39	40	3
Iris	4	3	3	12	50	3
Breast Cancer	30	3	3	90	40	2
Heart Disease	13	2	5	65	40	2
Parkinsons	22	2	3	66	40	2
Credit Approval	15	3	3	45	40	2
Pima Diabetes	8	3	3	24	40	2
BUPA	6	2	5	30	60	2
SAHeart	9	2	5	45	50	2
Haberman's	3	3	5	15	50	2
German Credit	20	3	5	100	50	2
WDBC	9	2	3	27	40	2

Çizelge 3.2'de çalışmada kullanılan veri setlerine göre oluşturulan yapay sinir ağı model mimarileri ve eğitim parametreleri gösterilmektedir. Yapay sinir ağı modellerinde giriş katmanındaki nöron sayısı (n), ilgili veri setindeki öznitelik sayısına göre belirlenmiştir. Çıkış katmanındaki nöron sayısı (k) ise veri setindeki sınıf sayısına bağlı olarak oluşturulmuştur. İkili sınıflandırma problemlerinde çıkış katmanı iki nörondan oluşurken, İris ve Wine veri setlerinde olduğu gibi çok sınıflı problemlerde üç çıkış nöronu kullanılmıştır. Çalışmada kullanılan gizli katman sayıları veri setlerinin yapısına göre iki veya üç katman olacak şekilde belirlenmiştir. Gizli katmanlardaki nöron sayıları ise giriş özellik sayısının belirli bir çarpan ile genişletilmesi prensibine dayanmaktadır. Bu kapsamda $n \times 3$ ve $n \times 5$ çarpanları kullanılmıştır. Örneğin Heart Disease veri setinde 13 giriş özelliği için $13 \times 5 = 65$ gizli nöron kullanılırken, Parkinsons veri setinde 22 giriş özelliği için $22 \times 3 = 66$ gizli nöron kullanılmıştır (Zama, 2023).

Sonuç olarak, oluşturulan yapay sinir ağı mimarisi, hem şifresiz ortamda yüksek doğruluk sağlayan hem de homomorfik şifreleme altında gerçekleştirilen şifreli çıkarım senaryolarında uygulanabilirliği olan bir yapı olarak tasarlanmıştır. Bu sayede, şifresiz ve şifreli çıkarım sonuçlarının adil ve tutarlı bir şekilde karşılaştırılması mümkün hale gelmiştir.

3.3.2. Eğitim parametreleri

Bu çalışmada kullanılan yapay sinir ağı modellerinin eğitim parametreleri, farklı veri setleri üzerinde elde edilen sınıflandırma doğruluğunu artırmak amacıyla deneysel olarak belirlenmiştir. Eğitim sürecinde, modelin genelleme yeteneğini koruyacak ve aşırı öğrenmeye neden olmayacak parametre kombinasyonları tercih edilmiştir.

Öğrenme oranı, modelin hata fonksiyonunu kararlı bir biçimde minimize edebilmesi için farklı değerler denenerek belirlenmiş; çok büyük değerlerde gözlemlenen kararsız yakınsama ve çok küçük değerlerde ortaya çıkan yavaş öğrenme problemi dikkate alınarak optimum bir değer seçilmiştir. Bu yaklaşım, modelin hem hızlı hem de istikrarlı bir şekilde eğitilmesini sağlamıştır.

Tüm modellerde doğrusal olmayan öğrenme yeteneği sağlayabilmek amacıyla ReLU (Rectified Linear Unit) aktivasyon fonksiyonu tercih edilmiştir. Eğitim süreçlerinde veri setlerinin yapısına bağlı olarak 40, 50 ve 60 epoch değerleri kullanılmıştır. Böylece farklı veri setlerinin boyut ve karmaşıklık düzeylerine uygun yapay sinir ağı mimarileri oluşturularak modellerin sınıflandırma performanslarının değerlendirilmesi amaçlanmıştır.

Modelin eğitim sürecinde kullanılan parametreler doğruluk performansını maksimize etmek ve eğitim süresini makul seviyede tutmak amacıyla deneysel olarak belirlenmiştir. Bu kapsamda, epoch sayısı, aktivasyon fonksiyonu ve model eğitim sürecine ilişkin temel ayarlar dikkate alınmıştır. Eğitim sürecinde maksimum epoch sayısı "max_epochs = 50" olarak belirlenmiştir. Yapılan deneysel değerlendirmelerde, 60 epoch ile elde edilen sonuçların 50 epoch ile elde edilen sonuçlara oldukça yakın olduğu, ancak eğitim süresinin belirgin şekilde arttığı gözlemlenmiştir. Bu nedenle, model performansı ile eğitim süresi arasında denge sağlamak amacıyla 50 epoch değeri tercih edilmiştir. Modelde aktivasyon fonksiyonu olarak ReLU (Rectified Linear Unit) kullanılmıştır. ReLU fonksiyonu, doğrusal olmayan ilişkilerin öğrenilmesini sağlarken aynı zamanda hesaplama açısından verimli bir yapı sunmaktadır. Bu özelliği sayesinde hem modelin eğitim sürecinde hem de daha sonra gerçekleştirilen şifreli çıkarım aşamasında uygun bir tercih olmuştur. Model eğitimi sırasında çıktıların sade tutulması amacıyla "verbose = 0" parametresi kullanılmıştır. Bu sayede eğitim süreci gereksiz detaylardan arındırılarak daha kontrollü bir şekilde yürütülmüştür. Ayrıca, modelin değerlendirilmesi amacıyla veri setleri %75 eğitim ve %25 test olacak şekilde ayrılmıştır. Bu ayrım sayesinde modelin genelleme performansı test verisi üzerinde objektif bir şekilde ölçülmüştür.

Optimizasyon yöntemi olarak gradyan tabanlı geri yayılım algoritması kullanılmış ve sınıflandırma problemlerine uygun bir kayıp fonksiyonu tercih edilmiştir. Bu seçim, farklı sınıflar arasındaki ayrımın daha etkili bir biçimde öğrenilmesine katkı sağlamıştır.

Batch boyutu, veri setlerinin örnek sayıları ve dağılımları dikkate alınarak belirlenmiştir. Çok küçük batch boyutlarının gradyan hesaplamalarında gürültüyü artırdığı, çok büyük batch boyutlarının ise genelleme performansını olumsuz etkilediği gözlemlendiğinden, bu iki durum arasında denge sağlayan bir batch boyutu kullanılmıştır.

Belirlenen eğitim parametreleri, tüm veri setleri için tutarlı bir şekilde uygulanmış ve şifresiz ortamda elde edilen modeller, Concrete-ML kullanılarak gerçekleştirilen şifreli çıkarım sürecinde doğrudan kullanılmıştır. Bu yaklaşım sayesinde, şifreli ve şifresiz senaryolar arasındaki performans farklarının eğitim sürecinden değil, homomorfik şifreleme mekanizmasından kaynaklandığı net bir biçimde analiz edilebilmiştir. Sonuç olarak, eğitim parametreleri belirlenirken yalnızca doğruluk oranı değil, aynı zamanda eğitim süresi ve model verimliliği de dikkate alınmış, bu doğrultuda dengeli bir yapı oluşturulmaya çalışılmıştır.

3.3.3. Şifresiz eğitim süreci

Bu çalışmada yapay sinir ağı modeli şifresiz ortamda eğitilmiştir. Eğitim sürecinde veri setleri, ön işleme adımlarının ardından eğitim ve test kümelerine ayrılmıştır. Model yalnızca eğitim verileri kullanılarak öğrenme sürecine tabi tutulmuş, test verileri eğitim aşamasında modele dahil edilmemiştir. Şifresiz modelin eğitim süreci, veri setlerinin eğitim ve test olarak ayrılması ile başlamaktadır. Bu çalışmada veri setleri %75 eğitim ve %25 test olacak şekilde bölünmüş ve sınıf dağılımının korunması amacıyla stratified örnekleme yöntemi kullanılmıştır.

Model eğitimi, Concrete-ML kütüphanesinin sağladığı benchmark fonksiyonu kullanılarak gerçekleştirilmiştir. Bu yöntem ile model hem Concrete-ML yapısı altında eğitilmiş hem de aynı anda karşılaştırma amacıyla bir scikit-learn modeli elde edilmiştir. Bu sayede, şifresiz ortamda çalışan referans model ile Concrete-ML modelinin performansı doğrudan karşılaştırılabilmiştir. Eğitilen modellerin performansı test verisi üzerinde değerlendirilmiştir. Bu kapsamda, scikit-learn modeli kullanılarak elde edilen tahminler ile gerçek etiketler karşılaştırılmış ve doğruluk değeri hesaplanmıştır. Aynı şekilde, Concrete-ML modelinin şifreleme uygulanmadan elde ettiği tahminler de değerlendirilmiştir. Elde edilen sonuçlar, şifresiz ortamda çalışan scikit-learn modeli ile

Concrete-ML modelinin benzer doğruluk değerlerine ulaştığını göstermektedir. Bu durum, Concrete-ML modelinin şifreleme uygulanmadan önce açık veri üzerinde doğru şekilde çalıştığını ve model davranışının korunduğunu ortaya koymaktadır.

Şifresiz modelden elde edilen bu sonuçlar, daha sonraki aşamada homomorfik şifreleme altında gerçekleştirilen model performansı ile karşılaştırılmak üzere referans olarak kullanılmıştır. Böylece, şifreleme sürecinin model doğruluğu üzerindeki etkisi açık bir şekilde analiz edilebilmiştir. Şifresiz eğitim sürecinin tercih edilme nedeni, güncel homomorfik şifreleme tabanlı makine öğrenmesi yaklaşımlarında model eğitiminin yüksek hesaplama maliyeti ve pratik kısıtlar nedeniyle genellikle şifresiz ortamda gerçekleştirilmesidir. Bu durum literatürde yaygın olarak benimsenmiş bir yaklaşımdır. Eğitim tamamlandıktan sonra elde edilen model, referans model olarak kullanılmış ve hem şifresiz test verileri hem de homomorfik şifreleme ile şifrelenmiş test verileri üzerinde çıkarım yapmak amacıyla kullanılmıştır. Böylece şifreli ve şifresiz çıkarım senaryoları arasında performans karşılaştırması yapılabilmiştir.

3.4. Concrete-ML ile Şifreli Modelleme

Bu çalışmada, homomorfik şifreleme altında makine öğrenmesi çıkarımı gerçekleştirmek amacıyla Zama tarafından geliştirilen Concrete-ML kütüphanesi kullanılmıştır. Concrete-ML, tam homomorfik şifreleme (Fully Homomorphic Encryption – FHE) tekniklerini kullanarak, şifrelenmiş veriler üzerinde makine öğrenmesi modelleri ile çıkarım yapılmasına olanak tanıyan açık kaynaklı bir kütüphanedir (Zama, 2023).

Çizelge 3.3 ‘de yapay sinir ağı ile torus homomorfik şifreleme deneysel süreç adımları algoritması verilmiştir.

Çizelge 3.3. Yapay sinir ağı ve homomorfik şifreleme deneysel süreç adımları algoritması

Algoritma: Deney Adımları
Girdi: *D - veri setleri kümesi* *M - yapay sinir ağı modeli* *H - hiperparametre kümesi* *S - normalizasyon yöntemleri* *FHE - Concrete-ML / TFHE altyapısı*
Çıktı: *R - performans karşılaştırma sonuçları* *T - hesaplama maliyeti sonuçları*

Çizelge 3.3.(devam) Yapay sinir ağı ve homomorfik şifreleme deneysel süreç adımları algoritması

Adım 1. *Veri setlerini yükle:*
Her veri seti Di ∈ D için yap
Di ← LoadDataset(Di)
Adım 2. *Veri ön işleme işlemlerini uygula:*
Eksik verileri kontrol et
Kategorik değişkenleri sayısal forma dönüştür
Giriş özellikleri X ve hedef değişken y olarak
Adım 3. *Veriyi eğitim ve test kümelerine ayır:*
X_train, X_test, y_train, y_test ← TrainTestSplit(X, y)
Adım 4. *Normalizasyon / ölçekleme işlemini uygula:*
Her ölçekleme yöntemi Sj ∈ S için tekrarla
X_train_scaled ← FitTransform(Sj, X_train)
X_test_scaled ← Transform(Sj, X_test)
Bitir döngü
Adım 5. *Şifresiz yapay sinir ağı modelini eğit:*
Her hiperparametre kombinasyonu h ∈ H için tekrarla
M_h ← CreateANNModel(h)
M_h ← Train(M_h, X_train_scaled, y_train)
y_pred_plain ← Predict(M_h, X_test_scaled)
Acc_plain ← Accuracy(y_test, y_pred_plain)
Adım 6. *En iyi modeli seç:*
M_best ← En yüksek doğruluklu (Max Acc_plain) model
Adım 7. *Eğitilmiş modeli Concrete-ML ortamına aktar:*
FHE_model ← ConvertToConcreteML(M_best)
Adım 8. *Modeli nicemle ve derle:*
Quantized_model ← Quantize(FHE_model)
Compiled_model ← Compile(Quantized_model, X_train_scaled)
Adım 9. *Test verisini TFHE ile şifrele:*
X_test_enc ← Encrypt(X_test_scaled)
Adım 10. *Şifreli veri üzerinde çıkarım yap:*
y_pred_enc ← Predict(Compiled_model, X_test_enc, fhe = "execute")
Adım 11. *Şifreli çıkarım sonucunu çöz:*
y_pred_fhe ← Decrypt(y_pred_enc)
Adım 12. *FHE model performansını hesapla:*
Acc_fhe ← Accuracy(y_test, y_pred_fhe)
Adım 13. *Şifresiz ve şifreli model sonuçlarını karşılaştır:*
Difference ← Acc_plain - Acc_fhe
R ← R ∪ {Di, Sj, M_best, Acc_plain, Acc_fhe, Difference}
Adım 14. *Hesaplama maliyetini kaydet:*
T_plain ← ExecutionTime(Predict(M_best, X_test_scaled))
T_fhe ← ExecutionTime(Predict(Compiled_model, X_test_enc))
T ← T ∪ {Di, T_plain, T_fhe}
Bitir döngü
Bitir döngü
Adım 15. *Tüm veri setleri için sonuçları tablo ve grafik olarak raporla:*
Çıktı (R, T)

3.4.1. Concrete-ML kütüphanesi ve TFHE altyapısı

Concrete-ML, geleneksel makine öğrenmesi modellerinin homomorfik şifreleme altında çalıştırılabilmesi için özel olarak tasarlanmıştır. Kütüphane, Python tabanlı olup Scikit-Learn ile uyumlu bir kullanım sunmaktadır. Bu sayede, şifresiz ortamda eğitilen

modeller minimal değişikliklerle homomorfik şifreleme altında çıkarım yapabilir hale getirilmektedir (Zama, 2023).

Bu çalışmada, sınıflandırma modellerinin şifrelenmiş veriler üzerinde doğrudan çalıştırılabilmesi için Concrete-ML kütüphanesi ile Torus tabanlı Tam Homomorfik Şifreleme (TFHE) altyapısı birlikte kullanılmıştır. Concrete-ML, makine öğrenmesi modellerini FHE çerçevesine dönüştürmek ve şifrelenmiş veriler üzerinde çıkarım yapmak için geliştirilmiş bir araç setidir. Bu kütüphane, veri bilimcilerinin kriptografi bilgisine sahip olmadan mevcut makine öğrenmesi modellerini FHE uyumlu hâle getirmesine imkân tanır; böylece modeller doğrudan şifreli girişler üzerinde tahmin gerçekleştirebilirler (Zama, 2023)

Tam Homomorfik Şifreleme (FHE), veri gizliliğini korurken şifrelenmiş veriler üzerinde aritmetik ve mantıksal işlemlerin yapılmasına imkân veren bir kriptografik tekniktir. TFHE (Torus Fully Homomorphic Encryption), FHE şemalarının bir çeşididir ve Torus yapısı ($\mathbb{R}/\mathbb{Z}$) üzerinde tanımlanır. TFHE, bit düzeyinde toplama ve çarpma işlemleri ile non-lineer fonksiyonların şifrelenmiş veriler üzerinde uygulanmasına olanak tanır. TFHE'nin temel avantajı, şifrelenmiş verilerde hesaplama yaparken güvenlik ve doğruluk sağlamasıdır (Chillotti vd., 2016).

Concrete-ML, TFHE altyapısını doğrudan kullanarak klasik makine öğrenmesi modellerini şifrelenmiş veri üzerinde çıkarım yapabilecek FHE eşdeğerlerine dönüştürür. Bu dönüşüm süreci üç aşamadan oluşmaktadır. İlk aşama model eğitimi (plaintext), sonraki aşama modelin FHE uyumlu hale getirilmesi ve son aşama şifrelenmiş veri üzerinde çıkarımlar yapılmıştır.

Concrete-ML, tahmin aşamasının şifreli veriler üzerinde gerçekleştirilmesine odaklanır ve bu aşamada verilerin açığa çıkmasını gerektirmez. Kütüphane, modelin tüm ağırlıklarını ve hesaplamalarını tamsayılara çevirerek FHE ile uyumlu hâle getirir; böylece TFHE altında çalışabilen bir çıkarım devresi oluşturur. Programmable bootstrapping gibi ileri teknikler sayesinde şifreli hesaplamaların doğruluğu artırılır (Zama, 2023).

Concrete-ML avantajları olduğu kadar dezavantajları da bulunmaktadır. Concrete-ML sayesinde veri her zaman şifreli tutulur ve model bu şifreli veriler üzerinde tahmin yapabilir. Bu yapı, özellikle gizlilik ve güvenlik gerektiren uygulamalarda örneğin finans, sağlık veya kişisel veri içeren diğer sınıflandırma problemleri önemli avantaj sağlar. Concrete-ML, TFHE'nin kriptografik gücünü makine öğrenmesi modelleriyle birleştirerek hem performans hem de gizlilik açısından pratik çözümler sunar.

3.4.2. TFHE şeması ve programmable bootstrapping (PBS)

TFHE (Torus Fully Homomorphic Encryption), bit düzeyinde işlemleri destekleyen ve Torus yapısı üzerinde tanımlanan bir Tam Homomorfik Şifreleme (FHE) şemasıdır. TFHE'nin temel amacı, şifrelenmiş veriler üzerinde lineer ve non-lineer fonksiyonları doğrudan uygulayabilmektir. Bu, özellikle yapay sinir ağları gibi non-lineer hesaplamalar gerektiren makine öğrenmesi modellerinin şifreli veriler üzerinde çalıştırılmasını mümkün kılar (Chillotti vd., 2016).

TFHE, klasik LWE (Learning With Errors) problemini Torus tabanlı bir temsil ile genişleterek, yüksek doğrulukta şifreleme ve çıkarım yapabilmeyi sağlar. Bit düzeyinde toplama ve çarpma işlemlerine izin verir, bu sayede activation function gibi non-lineer işlemler şifrelenmiş veriler üzerinde gerçekleştirilebilir.

TFHE'nin programmable bootstrapping (PBS) mekanizmasından bahsedelim. Bu mekanizma TFHE'nin en devrimsel bileşeni olan Programmable Bootstrapping (PBS), bootstrapping işlemini sadece pasif bir gürültü temizleme adımı olmaktan çıkarıp aktif bir hesaplama adımına dönüştürür. PBS, gürültü temizleme süreci sırasında şifreli metne herhangi bir tek değişkenli fonksiyonun (f(x)) uygulanmasına olanak tanır.

PBS mekanizmasının temel işlevleri arasında ilk olarak gürültü yöntemi ve sıfırlamadır. Bu kısımda Şifreli uzayda yapılan her aritmetik işlem sonucunda biriken gürültü seviyesi, deşifre edilebilir güvenli sınırlara geri çekilir. Diğer işlerleri arasınna fonksiyonel hesaplama (look-up table) kısmıdır. Bu kısımda ise PBS, hesaplanacak fonksiyonu bir "Look-Up Table (LUT)" olarak kullanır. Bootstrapping sırasında kullanılan test polinomu, bu tabloya göre döndürülerek (blind rotation) sonuç elde edilir. Bu sayede ReLU, Sigmoid veya Step gibi non-lineer aktivasyon fonksiyonları şifreli veriye ek maliyet getirmeden uygulanabilir (Chillotti vd., 2020).

Son işlevi ise sınırsız devre derinliğidir. Bu kısımda PBS sayesinde gürültü her adımda kontrol altına alındığı için, geleneksel FHE şemalarındaki sınırlı işlem derinliği problemi aşılır. Bu, çok katmanlı derin öğrenme modellerinin şifreli uzayda çalıştırılabilmesini mümkün kılar.

TFHE ve PBS'in sağladığı avantajlar bulunmaktadır. Veri ve model gizliliği çıkarım sürecinde veriler sunucu tarafında her zaman şifreli kalır; sunucu ne girdi verisini ne de modelin ürettiği ara sonuçları görebilir. Diğer bir avantajı kesin sonuçlar yaklaşık hesaplama yapan CKKS gibi şemaların aksine TFHE, nicemlenmiş tamsayılar üzerinde kesin sonuçlar üretir. Bu durum, modelin şifresiz uzaydaki doğruluğunun şifreli uzayda

da korunmasını sağlar. Başka bir avantajı da concrete-ML ile entegre edilen bu yapı, PBS işlemlerini CPU üzerinde optimize ederek özel bir kriptografik donanıma ihtiyaç duymadan makul sürelerde sonuç alınmasını sağlar.

Algoritma 3.4'te TFHE tabanlı torus şifreleme süreci adım adım gösterilmektedir. İlk aşamada açık metin mesajı torus uzayında temsil edilebilecek biçime dönüştürülmektedir. Ardından torus uzayından rastgele bir maske vektörü seçilmekte ve güvenliği sağlamak amacıyla Gauss dağılımına sahip küçük bir hata terimi üretilmektedir. Şifreleme işlemi sırasında açık metin, gizli anahtar ile ilişkili doğrusal işlem ve hata terimi ile birleştirilerek şifreli metin oluşturulmaktadır. Elde edilen (a, b) yapısı TFHE şemasındaki şifreli metni temsil etmektedir.

Çizelge 3.4. TFHE şifreleme süreci algoritması

Algoritma: TFHE Tabanlı Torus Şifreleme

Girdi: *m - açık metin mesajı*
$s \in \{0,1\}^n$ - gizli anahtar vektörü
σ - hata dağılımı parametresi

Çıktı: *ct = (a,b) - şifreli metin*

Adım 1. *Açık metin mesajını torus uzayına kodla:*
$\mu \leftarrow EncodeToTorus(m)$
Adım 2. *Torus uzayından rastgele maske vektörü seç:*
$a \leftarrow UniformRandom(\mathbb{T}^n)$
Adım 3. *Küçük hata terimi üret:*
$e \leftarrow GaussianNoise(0, \sigma)$
Adım 4. *Şifreli metnin ikinci bileşenini hesapla:*
$b \leftarrow \langle a, s \rangle + \mu + e \bmod 1$
Adım 5. *Şifreli metni oluştur:*
$ct \leftarrow (a,b)$
Adım 6. *ct değerini çıktı olarak ver.*

Bu yapı, Learning With Errors (LWE) problemine dayanan güvenlik yaklaşımını temel almakta olup şifreli veriler üzerinde doğrudan hesaplama yapılabilmesine olanak sağlamaktadır. Algoritmada kullanılan hata terimi (e), sistem güvenliğinin temel bileşenlerinden biri olarak görev yapmakta ve saldırganın gizli anahtarı elde etmesini zorlaştırmaktadır. Ayrıca torus tabanlı gösterim sayesinde TFHE şeması, programmable bootstrapping (PBS) mekanizması ile doğrusal olmayan işlemlerin şifreli veriler üzerinde gerçekleştirilebilmesini desteklemektedir. Bu nedenle TFHE, modern tam homomorfik şifreleme sistemleri içerisinde önemli bir yere sahiptir.

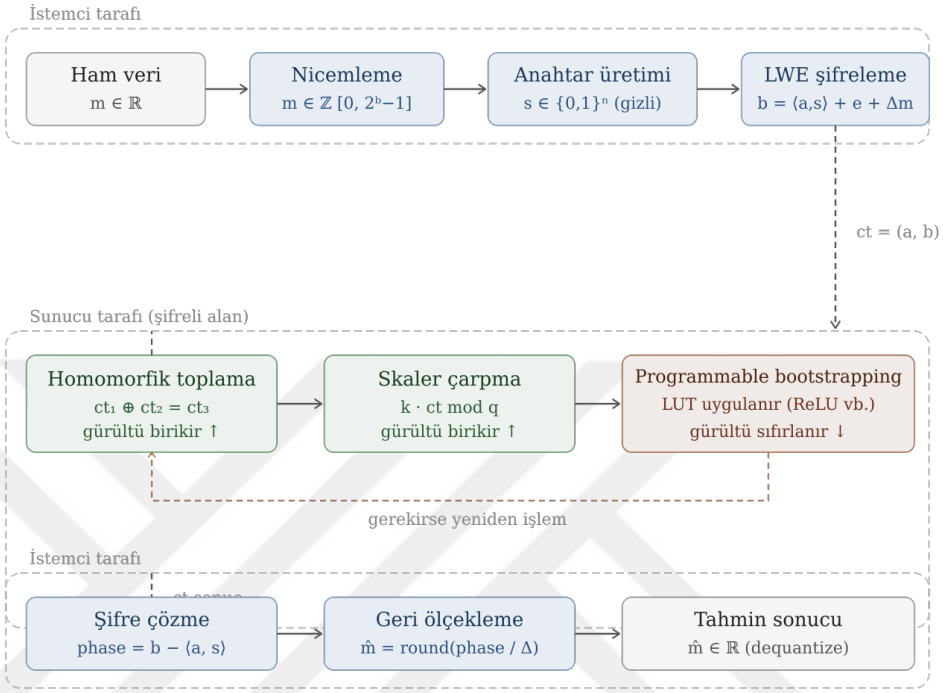
Bu çalışmada, homomorfik şifreleme sürecinin temel mantığını daha anlaşılır hale getirmek amacıyla, TFHE yaklaşımının bazı temel bileşenlerini kavramsal düzeyde yansıtan sadeleştirilmiş bir uygulama yapılmıştır. Yapılan homomorfik şifreleme

uygulamasına ait kodlar Ek 1'de sunulmuştur. Bu uygulama, gerçek üretim ortamlarında kullanılan TFHE sistemlerinin tüm matematiksel ve algoritmik ayrıntılarını içermemekle birlikte, yöntemin temel çalışma adımlarını açıklamak açısından öğretici bir yapı sunmaktadır. Kod içerisinde anahtar üretimi, LWE tabanlı şifreleme ve şifre çözme, homomorfik toplama, skaler çarpma, nicemleme, veri seti düzeyinde şifreleme ve sadeleştirilmiş programmable bootstrapping adımları tanımlanmıştır. Kodun ilk aşamasında şifreleme sistemi için kullanılacak parametreler belirlenmektedir. Bu parametreler arasında gizli anahtarın boyutu, modüler aritmetikte kullanılan alan, gürültü düzeyi ve düz metinlerin kaç bit ile temsil edileceği yer almaktadır. Özellikle "plaintext bits" parametresi, şifrelenecek verilerin kaç bitlik bir aralıkta temsil edileceğini belirlemektedir. Buna bağlı olarak delta adı verilen ölçekleme katsayısı hesaplanmakta ve düz metin değeri şifreli alanda uygun bir konuma taşınmaktadır. Bu işlem, mesajın modüler uzay içerisinde anlamlı bir biçimde temsil edilmesini sağlamaktadır.

İkinci aşamada gizli anahtar üretilmektedir. Kodda gizli anahtar, ikili değerlerden oluşan bir vektör olarak tanımlanmıştır. Bu vektör, LWE tabanlı şifreleme yapısının temelini oluşturmaktadır. Anahtar üretimi sırasında her bir eleman 0 veya 1 olacak şekilde rastgele seçilmekte ve böylece gizli anahtar vektörü elde edilmektedir. Bu anahtar hem şifreleme hem de şifre çözme işlemlerinde kullanılmaktadır. Şifreleme aşamasında, her bir düz metin değeri önce uygun aralıkta bir tam sayı olarak ele alınmaktadır. Daha sonra rastgele bir maske vektörü oluşturulmakta ve bu vektör ile gizli anahtar arasında bir iç çarpım hesaplanmaktadır. Bu işleme ek olarak sisteme rastgele bir hata terimi, yani gürültü eklenmektedir. Son olarak ölçeklenmiş mesaj değeri de bu yapıya dahil edilmektedir. Böylece şifreli metin, genel olarak (a,b) biçiminde ifade edilen iki bileşenli bir yapı olarak elde edilmektedir. Burada a rastgele seçilen maskeyi, b ise gizli anahtar ile yapılan iç çarpım, gürültü terimi ve mesaj bileşeninin birleşimini temsil etmektedir. Bu yapı, LWE tabanlı şifrelemenin temel formudur. Şifre çözme aşamasında ise şifreli metin içerisindeki a ve b bileşenleri kullanılarak, gizli anahtar yardımıyla mesaj geri elde edilmektedir. Bunun için önce b değerinden a ile gizli anahtarın iç çarpımı çıkarılmakta, ardından elde edilen sonuç ölçekleme katsayısına bölünerek en yakın tam sayıya yuvarlanmaktadır. Bu işlem sonunda düz metin değeri yaklaşık olarak geri kazanılmaktadır. Kodda bu süreç, ölçekli mesajın modüler alanda korunmasına dayanmaktadır. Kodun bir diğer önemli kısmı homomorfik işlemlerdir. Homomorfik şifreleme sistemlerinin temel özelliği, veriler şifreli haldeyken belirli matematiksel işlemlerin gerçekleştirilebilmesidir. Bu kodda iki temel işlem tanımlanmıştır:

homomorfik toplama ve skaler çarpma. Homomorfik toplama işleminde iki şifreli metin bileşen bazında toplanmakta ve sonuç yine şifreli bir metin olarak elde edilmektedir. Bu sayede, açık metinler üzerinde yapılacak toplama işlemi şifreli ortamda da gerçekleştirilebilmektedir. Benzer şekilde, bir şifreli metin sabit bir tam sayı ile çarpılarak skaler çarpma işlemi uygulanmaktadır. Bu işlem, özellikle doğrusal katmanların ve noktasal çarpım benzeri hesaplamaların şifreli alanda gerçekleştirilmesi açısından önemlidir. Kodda ayrıca nicemleme (quantization) adımı da yer almaktadır. Homomorfik şifreleme sistemleri, özellikle bu sadeleştirilmiş yapıda, doğrudan gerçek sayılarla değil tamsayılarla çalışmaktadır. Bu nedenle veri setindeki sürekli değerler önce belirli bir aralığa ölçeklendirilmekte ve daha sonra tamsayıya dönüştürülmektedir. Kodda bu işlem min-max yaklaşımına dayalı olarak gerçekleştirilmiştir. Verinin en küçük ve en büyük değerleri belirlenmekte, ardından tüm değerler belirli bir bit aralığına taşınarak tamsayı biçiminde temsil edilmektedir. Bu sayede veri, şifreleme işlemine uygun hale getirilmektedir. Nicemleme işleminden sonra veri seti düzeyinde şifreleme gerçekleştirilmektedir. Bu aşamada veri setindeki her bir örnek ve her bir özellik değeri ayrı ayrı ele alınmakta ve LWE tabanlı yöntemle şifrelenmektedir. Böylece veri seti, tamamı şifreli değerlerden oluşan bir yapıya dönüştürülmektedir. Bu yaklaşım, makine öğrenmesi modellerinde her bir giriş özelliğinin ayrı ayrı işlenebilmesine olanak sağlamaktadır. Şifre çözme aşamasında ise bu yapıdaki her değer tek tek çözülmekte ve sonrasında nicemleme öncesi yaklaşık gerçek değerler geri elde edilmektedir. Kodda yer alan önemli kavramlardan biri de programmable bootstrapping işlemidir. Gerçek TFHE sistemlerinde programmable bootstrapping, hem gürültüyü yenilemek hem de bir bakış tablosu (lookup table, LUT) aracılığıyla belirli bir fonksiyonu şifreli veri üzerinde uygulamak için kullanılmaktadır. Bu çalışmada yer alan kodda ise bu süreç sadeleştirilmiş biçimde modellenmiştir. Şifreli veri önce çözülmekte, ardından LUT yardımıyla yeni bir çıktı değeri elde edilmekte ve son olarak bu çıktı tekrar şifrelenmektedir. Her ne kadar bu yaklaşım gerçek TFHE sistemlerindeki ciphertext uzayında çalışan tam bootstrapping sürecini birebir temsil etmese de programmable bootstrapping kavramının temel amacını açıklamak açısından oldukça yararlıdır. Bu yöntem, özellikle ReLU gibi doğrusal olmayan fonksiyonların şifreli ortamda nasıl uygulanabileceğini kavramsal düzeyde göstermektedir. Kodun son kısmında homomorfik noktasal çarpım (dot product) işlemi yer almaktadır. Bu işlem, şifreli bir giriş vektörü ile düz metin biçimindeki ağırlıklar arasında gerçekleştirilmektedir. Her bir şifreli giriş değeri, ilgili ağırlık ile skaler olarak çarpılmakta ve ardından tüm sonuçlar homomorfik toplama ile birleştirilmektedir.

Böylece doğrusal bir katmanın temel işlemi olan ağırlıklı toplam hesaplanabilmektedir. Bu yapı, makine öğrenmesindeki doğrusal katmanların homomorfik şifreleme altında nasıl modellenebileceğini göstermesi açısından önemlidir.



Şekil 3.3. Homomorfik şifreleme (TFHE) sürecinin işlem adımları

Genel olarak değerlendirildiğinde bu Ekler 1 de bulunan kod, TFHE yaklaşımının temel mantığını adım adım açıklayan öğretici bir yapı sunmaktadır. Kodun amacı, gerçek üretim sistemlerinde kullanılan tüm düşük seviye optimizasyonları ve karmaşık matematiksel yapıları uygulamak değil; homomorfik şifreleme sürecinin ana bileşenlerini kavramsal olarak görünür hale getirmektir. Bu bağlamda anahtar üretimi, LWE tabanlı şifreleme, gürültü eklenmesi, homomorfik işlemler, nicemleme, veri seti şifreleme ve programmable bootstrapping kavramları aynı akış içinde bir araya getirilmiştir. Bu yapı, çalışmada kullanılan Concrete-ML tabanlı şifreli çıkarım sürecinin arka plan mantığını desteklemekte ve homomorfik şifrelemenin yapay sinir ağları üzerindeki uygulanışını daha anlaşılır hale getirmektedir.

3.4.3. Veri nicemleme (Quantization) ve şifreleme süreci

Bu çalışmada, veri gizliliğinin sağlanması amacıyla, çıkarım (inference) aşamasında kullanılacak giriş verileri homomorfik şifreleme yöntemi ile şifrelenmiştir. Şifreleme işlemi, bit düzeyinde hesaplamaları ve non-lineer fonksiyonları destekleyen TFHE (Torus Fully Homomorphic Encryption) şeması kullanılarak Concrete-ML kütüphanesi aracılığıyla gerçekleştirilmiştir. TFHE şeması, özellikle yapay sinir ağları gibi non-lineer ve hata toleranslı modellerin şifreli veriler üzerinde çalıştırılmasına uygunluğu nedeniyle tercih edilmiştir (Chillotti vd., 2016).

TFHE'de düz metin veriler, Torus yapısı üzerinde temsil edilir ve homomorfik işlemler bu yapı üzerinden gerçekleştirilir. TFHE'nin en önemli bileşeni olan Programmable Bootstrapping (PBS) mekanizması, şifreleme sırasında biriken gürültüyü temizlerken aynı zamanda non-lineer fonksiyonların (ReLU, Sigmoid, Step vb.) şifreli veriye doğrudan uygulanmasına imkân tanır. Bu yaklaşım, yapay sinir ağları gibi hata toleranslı sistemler için ideal bir çözüm sunar.

Ayrıca, Concrete-ML süreci kapsamında quantization (nicemleme) uygulanmıştır. Quantization, YSA modelinin ağırlık ve aktivasyon değerlerini TFHE'nin tamsayı tabanlı hesaplamalarına uygun biçimde dönüştürmek için kullanılır. Bu adım, şifreli çıkarım sırasında hesaplama maliyetini düşürür ve bellek kullanımını optimize eder. Quantization, verilerin veya model parametrelerinin doğruluk kaybını minimum seviyede tutarken, TFHE'nin bit düzeyinde çalışabilmesini mümkün kılar.

Bu çalışmada, ön işleme adımlarından geçirilmiş ve normalize edilmiş test verileri, Concrete-ML'nin derleme (compile) aşamasında oluşturulan kriptografik anahtarlar kullanılarak şifrelenmiştir. Concrete-ML, eğitilmiş modelin hesaplama grafiğini analiz ederek TFHE parametrelerini otomatik olarak belirler ve quantization ile şifreleme işlemini bu parametrelere uygun şekilde gerçekleştirir. Bu süreçte kullanıcıdan manuel parametre ayarı talep edilmemektedir.

Şifreleme işlemi tamamlandıktan sonra, veriler sunucu ortamına yalnızca şifreli biçimde aktarılmıştır. Şifreli veriler üzerinde gerçekleştirilen tüm hesaplamalar homomorfik olarak yapılmış olup, verinin açık hâline hiçbir aşamada erişilmemiştir. Bu yaklaşım, özellikle hassas verilerin üçüncü taraf sistemlerde işlenmesi senaryolarında veri gizliliğinin korunmasını sağlamaktadır (Zama, 2022).

Şifreli çıkarım sonucunda elde edilen çıktılar da şifreli biçimde üretilmiş ve yalnızca yetkili taraf tarafından şifre çözme işlemi uygulanarak anlamlı hâle getirilmiştir.

Böylece, giriş verileri hem de model çıktıları süreç boyunca gizli tutulmuştur. Bu yöntem, güncel literatürde TFHE tabanlı homomorfik şifreleme ile makine öğrenmesi uygulamalarında, quantization ile optimize edilmiş çıkarım senaryosu olarak yaygın biçimde kullanılmaktadır.

3.4.4. Şifreli çıkarım (Encrypted Inference)

Şifreli çıkarım, makine öğrenmesi modelinin şifreli veriler üzerinde çalıştırılarak tahmin üretmesi sürecini ifade etmektedir. Bu yaklaşımda model parametreleri şifresiz ortamda eğitilirken, modele verilen giriş verileri homomorfik şifreleme yöntemiyle TFHE kullanılarak şifrelenir. Böylece sunucu, verinin içeriğini görmeden hesaplama yapabilmekte ve veri gizliliği korunmaktadır (Gentry, 2009).

Bu çalışmada şifreli çıkarım işlemleri, TFHE (Torus Fully Homomorphic Encryption) şeması kullanılarak gerçekleştirilmiştir. TFHE, bit düzeyinde toplama ve çarpma işlemlerinin yanı sıra non-lineer fonksiyonları doğrudan şifreli veriye uygulayabilme yeteneği sayesinde makine öğrenmesi uygulamaları için uygun bir yapı sunmaktadır (Chillotti vd., 2016). Özellikle yapay sinir ağlarında kullanılan doğrusal ve doğrusal olmayan işlemler, TFHE'nin sunduğu Programmable Bootstrapping (PBS) mekanizması ile homomorfik ortamda uygulanabilmektedir.

Concrete-ML kütüphanesi, eğitilmiş yapay sinir ağı modellerinin TFHE altında çalıştırılabilmesini sağlayan bir çerçeve sunmaktadır. Bu kütüphane, şifreli çıkarım sürecinde modelin aktivasyon fonksiyonlarını ve diğer hesaplamalarını TFHE ile uyumlu hâle getirerek, PBS mekanizması ve quantization tekniklerini kullanır. Böylece şifreli ortamda model çıktıları güvenli ve doğru bir şekilde hesaplanabilmektedir (Zama, 2022).

Bu tez kapsamında, kullanıcıya ait test verileri önce TFHE kullanılarak şifrelenmiş, ardından eğitilmiş yapay sinir ağı modeli bu şifreli veriler üzerinde çıkarım yapmıştır. Sunucu tarafında gerçekleştirilen tüm hesaplamalar şifreli olarak yürütülmüş ve elde edilen tahmin sonuçları yine şifreli biçimde kullanıcıya iletilmiştir. Kullanıcı, yalnızca kendisine ait gizli anahtar ile sonuçları çözerek tahmin değerlerine erişebilmiştir.

Şifreli çıkarım yaklaşımı sayesinde, özellikle sağlık verileri gibi hassas verilerin gizliliği korunurken, merkezi bir sunucu üzerinde makine öğrenmesi hizmeti sunulabilmektedir. Bu yöntem, veri sahibinin verisini üçüncü taraflarla paylaşmadan analiz yaptırabilmesine olanak tanımakta ve güvenli yapay zekâ uygulamalarının temelini oluşturmaktadır.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu bölümde, tamamen homomorfik şifreleme kullanılarak gerçekleştirilen şifreli çıkarım işlemlerinin performansı değerlendirilmiş ve elde edilen sonuçlar şifresiz senaryolar ile karşılaştırmalı olarak incelenmiştir. Çalışma kapsamında farklı veri setleri üzerinde eğitilen yapay sinir ağı modellerinin hem şifresiz ortamda hem de tam homomorfik şifrelemenin alt şemalarından TFHE şifreleme altında elde edilen sonuçları analiz edilmiştir. Ayrıca, şifreleme kullanımının doğruluk ve hesaplama maliyeti üzerindeki etkileri tartışılmıştır.

4.1. Performans Değerlendirme Ölçütleri

Çalışma kapsamında toplam 12 farklı veri seti kullanılmıştır. Veri setleri; örnek sayısı, öznitelik sayısı ve problem zorluğu bakımından heterojen bir yapı sunmaktadır. Her veri seti için model eğitimi, homomorfik şifreleme ile uyumlu model dönüşümü ve şifreli çıkarım hesaplanmıştır. Z-Score normalizasyonu, Min-Max ölçekleme [0,1] ve [-1,1] ayrı ayrı uygulanmıştır.

Bu çalışmada modellerin doğruluk oranı (performansı, accuracy) ve hesaplama süresi saniye cinsinden (execution time) ölçülmüştür. Sklearn modeli, klasik yöntemde yani yapay sinir ağı modeli için kullanılmıştır. Concrete-ML, tam homomorfik şifreleme ile uyumlu olduğu için tercih edilmiştir.

TFHE şeması, bit düzeyinde toplama ve çarpma işlemlerinin yanı sıra non-lineer fonksiyonları doğrudan şifreli veriye uygulayabilme yeteneği sayesinde, reel sayılar üzerinde gerçekleştirilecek işlemleri güvenli bir şekilde şifreli alan içerisinde yürütmeye olanak tanır. Bu özellik, özellikle yapay sinir ağları gibi sürekli değerlerle çalışan makine öğrenmesi modelleri için TFHE'yi uygun bir şifreleme yöntemi hâline getirmektedir. Bu nedenle, Concrete-ML kütüphanesi ile FHE tabanlı model uygulamalarında TFHE tabanlı homomorfik şifreleme kullanılmıştır.

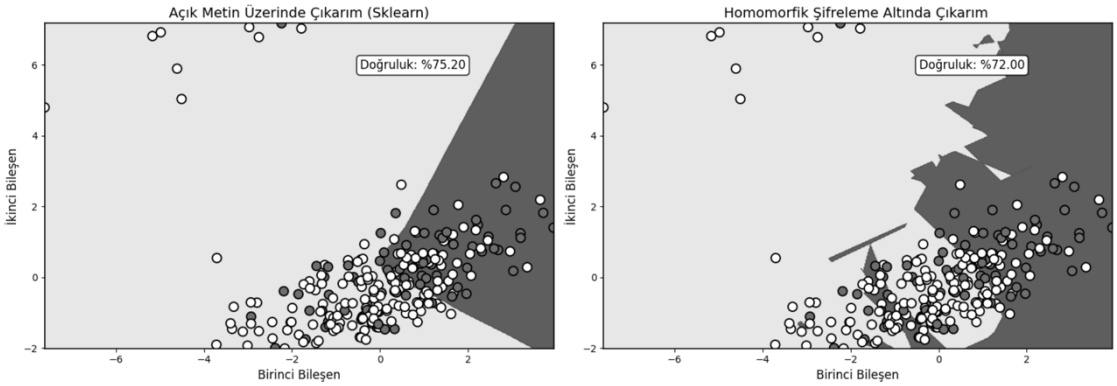
4.2. Şifresiz Model ve FHE Model Karşılaştırması

Çalışmanın bu kısmında veri setlerini inceleyeceğiz. Veri setlerine şifresiz model ile şifreleme yapılmış şekilde sonuçları karşılaştıracağız.

German Credit veri seti, bireylerin finansal durumları, gelir düzeyleri, kredi geçmişleri ve kişisel bilgileri gibi son derece hassas verileri içermektedir. Bu tür verilerin açık (plaintext) halde işlenmesi, yetkisiz erişim, veri ihlalleri ve kötüye kullanım gibi

ciddi güvenlik risklerini beraberinde getirmektedir. Özellikle finans sektöründe, müşteri verilerinin gizliliğinin korunması hem yasal düzenlemeler hem de etik sorumluluklar açısından büyük önem taşımaktadır. Geleneksel makine öğrenmesi yaklaşımlarında, verilerin işlenebilmesi için şifrelerinin çözülmesi gerekmektedir. Ancak bu durum, verilerin işlem sürecinde açığa çıkmasına ve potansiyel güvenlik açıklarının oluşmasına neden olmaktadır. Bu bağlamda, verilerin şifreleri çözülmeden işlenebilmesi büyük bir gereklilik olarak ortaya çıkmaktadır. Bu çalışmada kullanılan homomorfik şifreleme yaklaşımı, verilerin şifreli haldeyken dahi işlenebilmesine olanak tanımaktadır. Böylece hassas finansal veriler, gizliliği korunarak analiz edilebilmekte ve makine öğrenmesi modelleri güvenli bir şekilde uygulanabilmektedir. Bu özellik, özellikle kredi değerlendirme sistemleri gibi kritik uygulamalarda büyük bir avantaj sağlamaktadır. Bu veri seti üzerinde, hem şifresiz (plaintext) yapay sinir ağı modeli hem de homomorfik şifreleme (TFHE) tabanlı model uygulanmıştır. Elde edilen sonuçların karşılaştırılması amacıyla, her iki model için karar sınırlarını gösteren grafikler oluşturulmuştur.

Aşağıda verilen grafiklerde, sol tarafta şifresiz modelin, sağ tarafta ise şifreli modelin sonuçları yer almaktadır. Grafikler, modelin farklı sınıfları nasıl ayırdığını görsel olarak ortaya koymakta ve şifreleme işleminin model performansı üzerindeki etkisini değerlendirmeye olanak tanımaktadır.

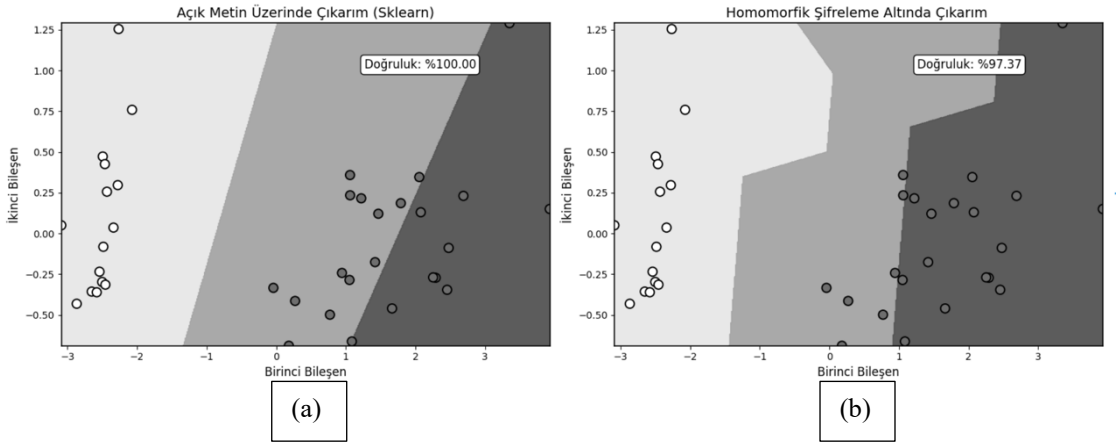


Şekil 4.1. German Credit veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması

Şekil 4.1'de, German Credit veri seti için şifresiz yani açık metin üzerinde (Sklearn) model ile homomorfik şifreleme (Concrete ML-TFHE) tabanlı modelin karar sınırları karşılaştırmalı olarak sunulmuştur. Elde edilen sonuçlara göre, açık veri üzerinde çalışan model %75,20 doğruluk oranı elde ederken, homomorfik şifreleme altında çalışan

model %72,00 doğruluk oranına ulaşmıştır. Bu durum, şifreleme ve nicemleme (quantization) süreçlerinin model performansında yaklaşık %3,20 oranında bir azalmaya neden olduğunu göstermektedir. Grafikler incelendiğinde, açık veri üzerinde çalışan modelin karar sınırlarının daha keskin ve belirgin bir yapı sergilediği görülmektedir. Buna karşılık, homomorfik şifreleme altında çalışan modelin karar sınırlarında, şifreli uzayda gerçekleştirilen hesaplamaların doğası gereği daha pürüzlü ve parçalı geçişler oluştuğu gözlemlenmektedir. Bu farklılıkların temel nedeni, TFHE tabanlı sistemlerde kullanılan nicemleme işlemleri ve şifreleme sürecinde oluşan gürültü (noise) etkisidir. Buna rağmen, homomorfik şifreleme tabanlı modelin veri dağılımının genel yapısını büyük ölçüde koruduğu ve sınıflandırma performansını kabul edilebilir bir seviyede sürdürdüğü görülmektedir. Elde edilen bulgular, homomorfik şifreleme yöntemlerinin veri gizliliğini koruyarak güvenli makine öğrenmesi uygulamalarına olanak sağladığını ve bunun sınırlı bir performans kaybı ile gerçekleştirilebildiğini ortaya koymaktadır.

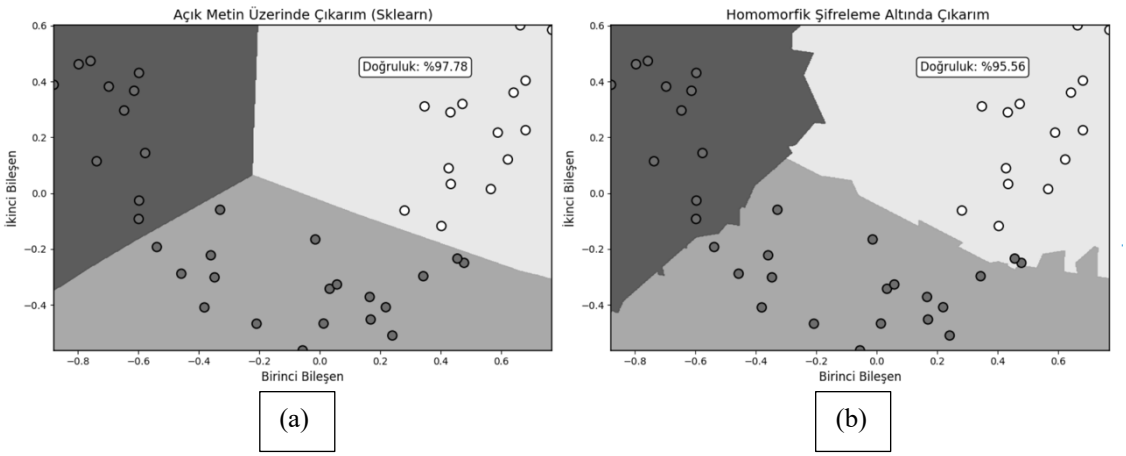
İris veri seti, homomorfik şifreleme yöntemlerinin makine öğrenmesi üzerindeki etkisini gözlemlemek amacıyla kullanılmıştır. Veri setinin düşük boyutlu ve dengeli yapısı, şifrelenmiş veri üzerinde gerçekleştirilen işlemlerin doğruluğunu analiz etmek için uygun bir ortam sunmaktadır. Şifrelenmiş veriler üzerinde doğrudan işlem yapılabilmesini sağlayan homomorfik şifreleme yaklaşımı sayesinde, veri gizliliği korunurken aynı zamanda makine öğrenmesi modellerinin uygulanabilirliği değerlendirilebilmektedir. Bu bağlamda, iris veri seti üzerinde hem şifresiz hem de TFHE tabanlı şifreli model uygulanarak karşılaştırmalı analiz yapılmıştır.



Şekil 4.2. İris veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması

Şekil 4.2'de, İris veri seti için açık yani şifresiz veri üzerinde eğitilen Sklearn modeli ile tam homomorfik şifreleme (FHE) kapsamında Concrete ML (TFHE) tabanlı modelin karar sınırları karşılaştırmalı olarak sunulmuştur. Elde edilen sonuçlara göre, açık veri üzerinde çalışan model %100,00 doğruluk oranı elde ederken, homomorfik şifreleme altında çalışan model %97,37 doğruluk oranına ulaşmıştır. Bu durum, şifreleme ve nicemleme (quantization) süreçlerinin model performansında yaklaşık %2,63 oranında bir azalmaya neden olduğunu göstermektedir. Grafikler incelendiğinde, açık veri üzerinde çalışan modelin karar sınırlarının oldukça net ve belirgin olduğu görülmektedir. Buna karşılık, homomorfik şifreleme altında çalışan modelin karar sınırlarında, şifreli uzayda gerçekleştirilen hesaplamaların doğası gereği sınırlı düzeyde sapmalar ve daha pürüzlü geçişler oluştuğu gözlemlenmektedir. Bu farklılıkların temel nedeni, TFHE tabanlı sistemlerde kullanılan nicemleme işlemleri ve şifreleme sürecinde ortaya çıkan gürültü (noise) etkisidir. Buna rağmen, homomorfik şifreleme tabanlı modelin veri noktalarını büyük ölçüde doğru bir şekilde ayırabildiği ve sınıflandırma performansını yüksek seviyede koruduğu görülmektedir. Elde edilen bulgular, homomorfik şifreleme yöntemlerinin yüksek doğruluk gerektiren veri setlerinde dahi veri gizliliğini koruyarak etkili sonuçlar üretebildiğini ve bunun düşük seviyede bir performans kaybı ile mümkün olduğunu ortaya koymaktadır.

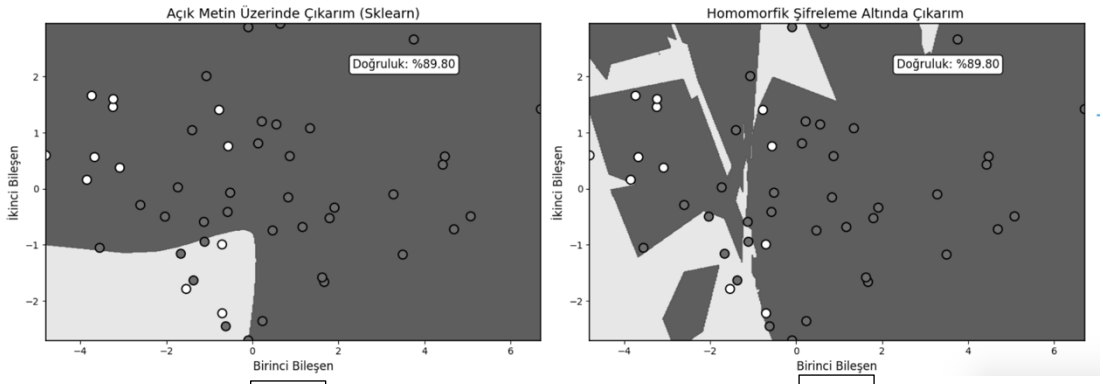
Wine veri seti üzerinde hem şifresiz yapay sinir ağı modeli hem de TFHE tabanlı şifreli model uygulanmıştır. Böylece, homomorfik şifreleme sürecinin model doğruluğu ve karar mekanizması üzerindeki etkisi analiz edilmiştir.



Şekil 4.3. Wine veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması

Şekil 4.3’te, şifresiz model ile homomorfik şifreleme kullanılarak elde edilen modelin karar sınırları karşılaştırmalı olarak sunulmuştur. Grafikler incelendiğinde, şifresiz modelin daha düzgün ve belirgin karar sınırları oluşturduğu görülmektedir. Homomorfik şifreleme (FHE) ile elde edilen modelde ise karar sınırlarının daha parçalı ve keskin geçişler içerdiği gözlemlenmektedir. Bu durum, şifreleme sürecinde uygulanan nicemleme (quantization) ve hesaplama sınırlamalarından kaynaklanmaktadır. Doğruluk değerleri karşılaştırıldığında, şifresiz model %97,78 doğruluk elde ederken, FHE modeli %95,56 doğruluk sağlamıştır. Bu fark oldukça düşük olup, homomorfik şifrelemenin model performansını büyük ölçüde koruduğunu göstermektedir. Ayrıca, Wine veri setinde elde edilen yüksek doğruluk değerleri, veri setinin sınıflarının belirgin şekilde ayrılabilir olduğunu ve modelin bu ayrımı başarılı bir şekilde öğrenebildiğini göstermektedir. Sonuç olarak, elde edilen bulgular homomorfik şifreleme yöntemlerinin, doğrulukta minimal bir kayıp ile güvenli makine öğrenmesi uygulamalarında etkin bir şekilde kullanılabileceğini ortaya koymaktadır.

Parkinson veri seti, homomorfik şifreleme yöntemi kullanılarak verilerin şifreli halde işlenmesi sağlanmış ve böylece veri gizliliği korunarak makine öğrenmesi uygulaması gerçekleştirilmiştir. Bu yaklaşım, özellikle sağlık verilerinin güvenli bir şekilde analiz edilmesi açısından önemli bir avantaj sunmaktadır. Parkinson veri seti üzerinde gerçekleştirilen deneyler sonucunda elde edilen doğruluk değerleri bulunmuştur.



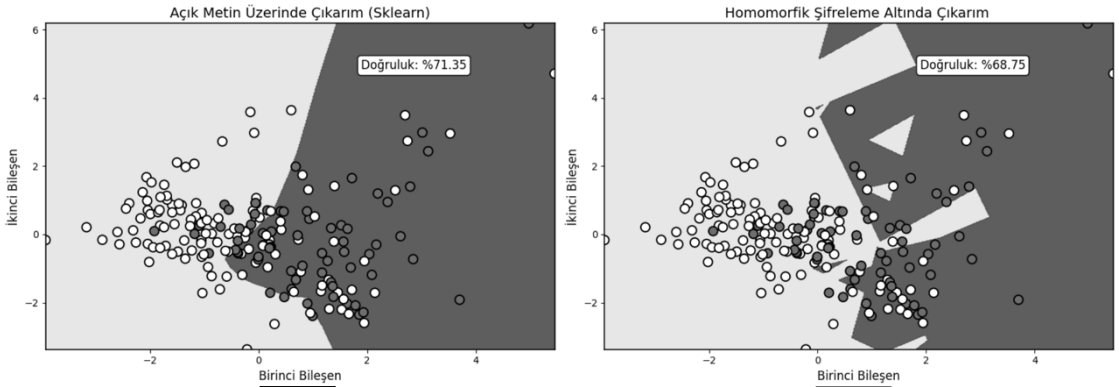
(a) (b)

Şekil 4.4. Parkinson veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması

Şekil 4.4’te, şifresiz model ile homomorfik şifreleme kullanılarak elde edilen modelin karar sınırları karşılaştırmalı olarak sunulmuştur. Grafikler incelendiğinde, şifresiz model ile homomorfik şifreleme kullanılarak elde edilen modelin karar sınırlarının büyük ölçüde benzer olduğu görülmektedir. Her iki modelin de veri dağılımını

benzer şekilde öğrendiği ve sınıfları ayırt etme konusunda aynı performansı sergilediği anlaşılmaktadır. Doğruluk değerlerinin tamamen aynı olması (89,80), homomorfik şifreleme sürecinin model performansını etkilemediğini açık bir şekilde ortaya koymaktadır. Bu durum, şifrelenmiş veriler üzerinde gerçekleştirilen hesaplamaların doğruluğunun, şifresiz ortamda elde edilen sonuçlarla tutarlı olduğunu göstermektedir. Ayrıca, simüle edilen model ile gerçek FHE modelinin aynı sonuçları üretmesi, sistemin doğru şekilde çalıştığını ve şifreleme sürecinin güvenilir olduğunu göstermektedir. Sonuç olarak, Parkinson veri seti üzerinde elde edilen bulgular, homomorfik şifreleme yöntemlerinin sağlık verileri gibi hassas veriler üzerinde doğruluk kaybı olmadan uygulanabileceğini ve güvenli makine öğrenmesi için güçlü bir yaklaşım sunduğunu ortaya koymaktadır.

Pima Indians Diabetes veri seti üzerinde elde edilen doğruluk sonuçları bulunmuştur. Şekil 4,5'te Şifresiz model doğruluğu %71,35 bulunurken homomorfik şifreleme ile şifrelenmiş model doğruluğu %68,75 olduğu saptanmıştır. Bu durum, şifreleme ve nicemleme (quantization) süreçlerinin model performansında yaklaşık %2,60 oranında bir azalmaya neden olduğunu göstermektedir.

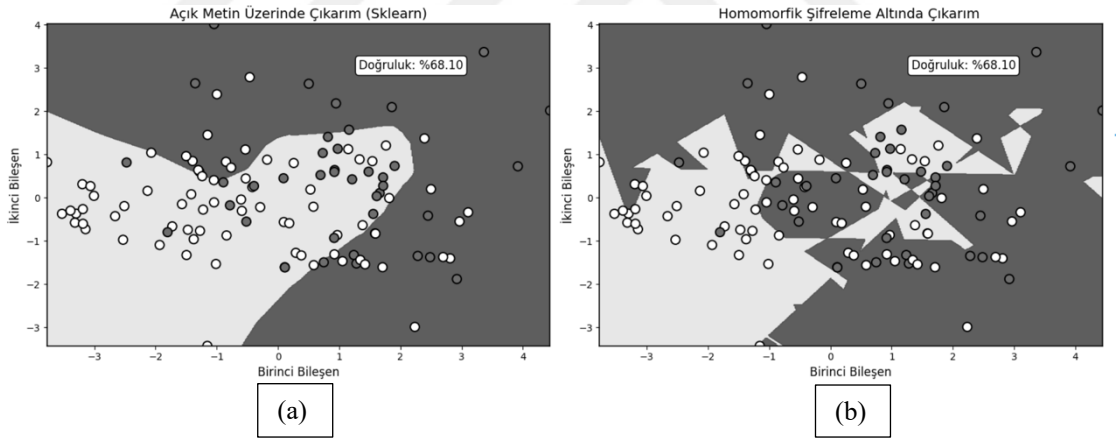


Şekil 4.5. Pima Indias Diabetes veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması

Grafikler incelendiğinde, açık veri üzerinde çalışan modelin karar sınırlarının daha dengeli ve veri dağılımına daha uyumlu olduğu görülmektedir. Buna karşılık, homomorfik şifreleme altında çalışan modelin karar sınırlarında daha düzensiz, parçalı ve yer yer keskin geçişlerin oluştuğu gözlemlenmektedir. Bu durum, özellikle veri setinin doğası gereği sınıflar arasında belirgin bir ayrımın bulunmaması ve örneklerin örtüşen dağılımlar göstermesi ile birlikte değerlendirildiğinde daha belirgin hale gelmektedir.

Pima Indians Diabetes veri seti, gerçek dünya problemlerine yakın yapısı ve sınıflar arasındaki ayrımın zor olması nedeniyle, model performansının sınırlı olduğu veri setlerinden biridir. Bu nedenle, homomorfik şifreleme sürecinin etkileri bu veri setinde daha belirgin şekilde ortaya çıkmaktadır. Ancak buna rağmen, FHE tabanlı modelin genel sınıflandırma performansını büyük ölçüde koruduğu ve açık veri modeline yakın sonuçlar üretebildiği görülmektedir. Elde edilen bulgular, veri setinin karmaşıklığı ve sınıf ayrımının zorluğu arttıkça, homomorfik şifreleme kaynaklı performans kayıplarının daha belirgin hale geldiğini göstermektedir. Bununla birlikte, elde edilen sonuçlar homomorfik şifreleme yöntemlerinin gerçek dünya verileri üzerinde de kabul edilebilir doğruluk seviyeleri ile güvenli makine öğrenmesi uygulamalarına olanak sağladığını ortaya koymaktadır.

SAHeart veri seti sağlık verilerinin güvenli analizi açısından önemli bir avantaj sağlamaktadır. SAHeart veri seti üzerinde gerçekleştirilen deneyler sonucunda Şekil 4.6'da şifresiz model ile homomorfik şifreleme kullanılarak elde edilen modelin karar sınırları karşılaştırmalı olarak gösterilmektedir.

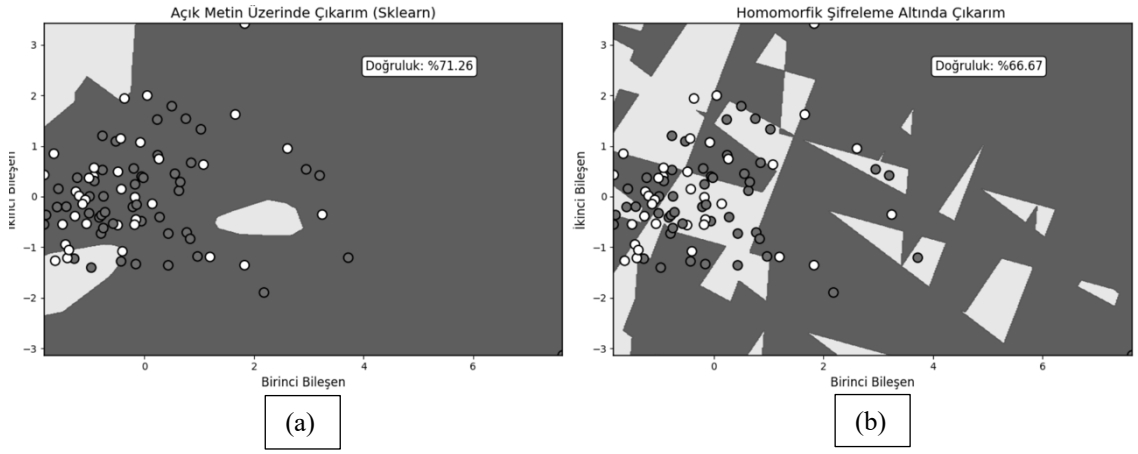


Şekil 4.6. SAHeart veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması

Grafikler incelendiğinde, SAHeart veri seti için açık veri üzerinde eğitilen Sklearn modeli ile tam homomorfik şifreleme kapsamında Concrete ML (TFHE) tabanlı modelin karar sınırları karşılaştırmalı olarak sunulmuştur. Elde edilen sonuçlara göre, her iki model de %68,10 doğruluk oranı elde etmiş olup, şifreleme sürecinin genel doğruluk metriği üzerinde belirgin bir etkisi olmadığı görülmektedir. Bununla birlikte grafikler incelendiğinde, açık veri üzerinde çalışan modelin karar sınırlarının daha tutarlı ve veri dağılımına uyumlu olduğu; homomorfik şifreleme altında çalışan modelin karar

sınırlarının ise oldukça parçalı, düzensiz ve yer yer keskin kırılmalar içerdiği dikkat çekmektedir. Bu durum, şifreli uzayda gerçekleştirilen hesaplamaların doğası gereği ortaya çıkan nicemleme (quantization) ve gürültü (noise) etkilerinin karar sınırlarını önemli ölçüde bozduğunu göstermektedir. SAHeart veri setinin sınıflar arasında belirgin bir ayrım sunmayan ve örtüşen örnekler içeren yapısı göz önüne alındığında, bu tür düzensizliklerin sınıflandırma doğruluğuna doğrudan yansımaması mümkündür. Nitekim, her iki modelin aynı doğruluk değerine ulaşması, farklı karar sınırı geometrilerine rağmen benzer sınıflandırma sonuçlarının elde edilebildiğini göstermektedir. Elde edilen bulgular, homomorfik şifreleme yöntemlerinin doğruluk açısından istikrarlı sonuçlar sunabildiğini; ancak modelin karar sınırı yapısı ve yorumlanması üzerinde belirgin değişimlere neden olabileceğini ortaya koymaktadır. Bu durum, özellikle sağlık verileri gibi hassas alanlarda model davranışının yalnızca doğruluk metriği ile değil, karar sınırlarının yapısal özellikleri ile değerlendirilmesi gerektiğini göstermektedir.

BUPA veri seti, homomorfik şifreleme yöntemi kullanılarak veriler şifreli biçimde işlenmiş ve böylece bireylerin sağlık bilgileri açığa çıkarmamak için Şekil 4.7'de, şifresiz model ile homomorfik şifreleme kullanılarak elde edilen modelin karar sınırları karşılaştırmalı olarak sunulmaktadır.

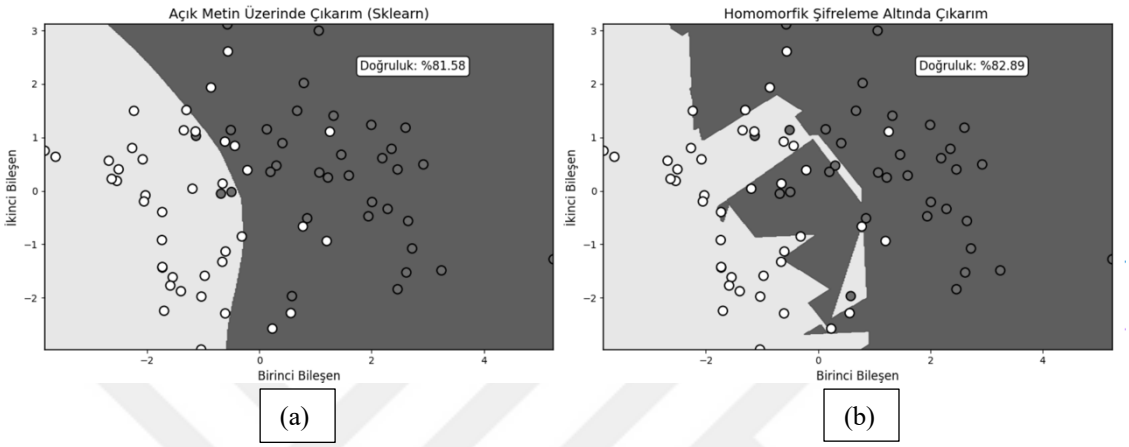


Şekil 4.7. BUPA veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması

Şekil 4.7 incelendiğinde, BUPA veri seti için açık veri üzerinde eğitilen Sklearn modeli ile tam homomorfik şifreleme (FHE) kapsamında Concrete ML (TFHE) tabanlı modelin karar sınırları karşılaştırmalı olarak sunulmuştur. Elde edilen sonuçlara göre, açık veri üzerinde çalışan model %71,26 doğruluk oranı elde ederken, homomorfik

şifreleme altında çalışan model %66,67 doğruluk oranına ulaşmıştır. Bu durum, şifreleme ve nicemleme (quantization) süreçlerinin model performansında yaklaşık %4,59 oranında bir azalmaya neden olduğunu göstermektedir.

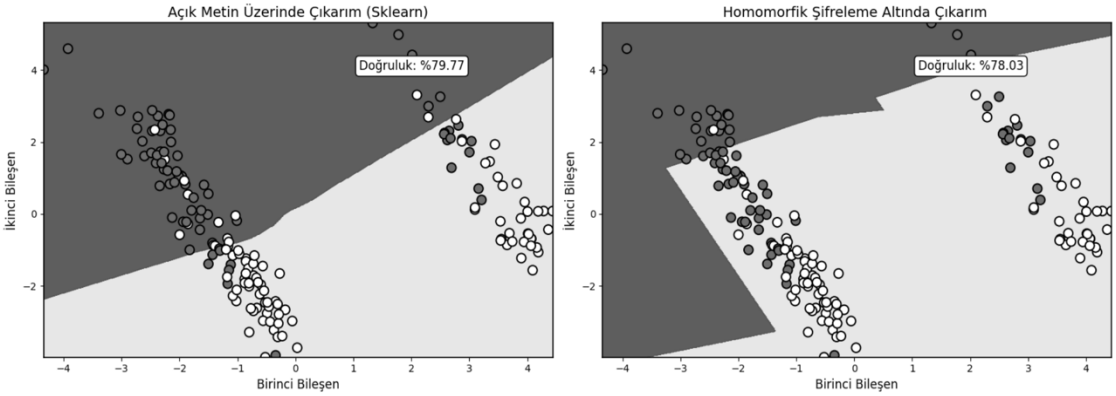
Heart Disease veri seti için Şekil 4.8'de, şifresiz model ile homomorfik şifreleme kullanılarak elde edilen modelin karar sınırları karşılaştırmalı olarak sunulmaktadır.



Şekil 4.8. Heart Disease veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması

Şekil 4.8 incelendiğinde, şifresiz model ile şifreli modelin karar sınırlarının genel olarak benzer olduğu, ancak şifreli modelde karar sınırlarının daha keskin ve parçalı hale geldiği gözlemlenmektedir. Bu durum, nicemleme (quantization) işleminin etkisiyle açıklanabilmektedir. Doğruluk değerleri incelendiğinde, şifreli modelin doğruluğunun (%86,84) şifresiz modele (%81,58) göre daha yüksek olduğu görülmektedir. Bu durum, homomorfik şifreleme sürecinin bazı veri setlerinde model performansını artırabileceğini göstermektedir. Bu artışın temel nedeni, nicemleme işleminin model üzerinde düzenleyici (regularization) bir etki oluşturarak aşırı öğrenmeyi azaltması olabilir. Özellikle orta düzey karmaşıklığa sahip veri setlerinde bu etkinin daha belirgin olduğu görülmektedir. Simüle edilen model ile gerçek FHE modelinin aynı doğruluk değerine sahip olması, şifreleme sürecinin doğru şekilde uygulandığını ve sonuçların güvenilir olduğunu göstermektedir. Sonuç olarak, Heart Disease veri seti üzerinde elde edilen bulgular, homomorfik şifreleme yöntemlerinin yalnızca veri gizliliği sağlamakla kalmayıp, bazı durumlarda model performansını da artırabileceğini ortaya koymaktadır.

Credit Approval veri seti için Şekil 4.9'da şifresiz model ile homomorfik şifreleme kullanılarak elde edilen modelin karar sınırları karşılaştırmalı olarak gösterilmektedir.

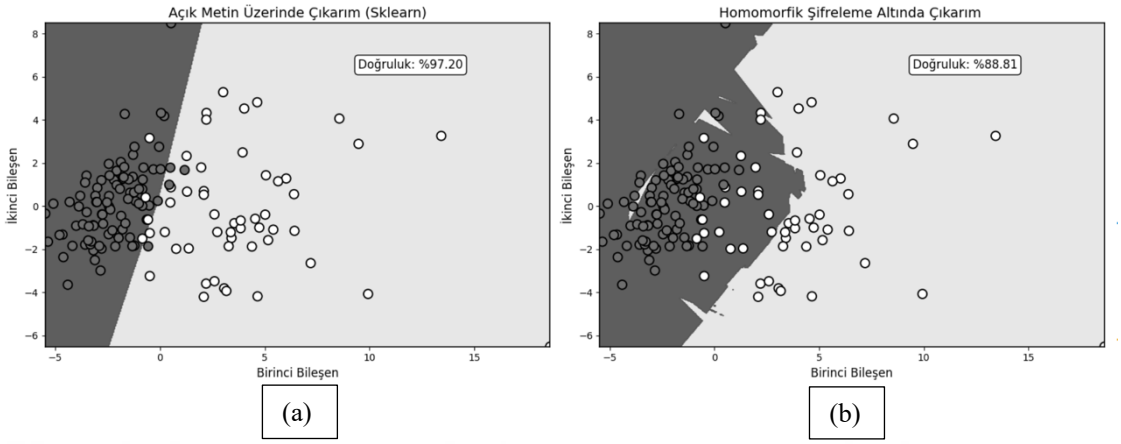


Şekil 4.9. Credit Approval veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması

Şekil 4.9' da (a) ve (b) her iki grafik incelendiğinde, şifresiz modelin daha düzgün ve dengeli karar sınırları oluşturduğu, buna karşılık şifreli modelde karar sınırlarının daha keskin ve doğrusal hale geldiği görülmektedir. Elde edilen sonuçlara göre, açık veri üzerinde çalışan model %79,77 doğruluk oranı elde ederken, homomorfik şifreleme altında çalışan model %78,03 doğruluk oranına ulaşmıştır. Bu durum, şifreleme ve nicemleme (quantization) süreçlerinin model performansında yaklaşık %1,74 oranında sınırlı bir azalmaya neden olduğunu göstermektedir. Açık veri üzerinde çalışan modelin karar sınırlarının daha düzgün ve veri dağılımına uyumlu olduğu görülmektedir. Homomorfik şifreleme altında çalışan modelin karar sınırlarında ise yer yer küçük düzensizlikler ve kırılmalar gözlemlenmektedir. Ancak bu farklılıkların genel karar yapısını önemli ölçüde bozmadığı ve sınıflar arasındaki ayrımın büyük ölçüde korunduğu dikkat çekmektedir. Credit Approval veri setinde sınıflar arasındaki ayrımın belirli bir ölçüde sağlanabilir olması, homomorfik şifreleme sürecinin etkilerinin sınırlı kalmasına katkı sağlamıştır. Bu durum, veri setinin ayrıştırılabilirliğinin yüksek olduğu senaryolarda, FHE tabanlı modellerin açık veri modellerine oldukça yakın performans sergileyebildiğini göstermektedir. Elde edilen bulgular, homomorfik şifreleme yöntemlerinin yalnızca yüksek doğruluklu veri setlerinde değil, orta seviyede ayrıştırılabilirliğe sahip veri setlerinde de güvenilir sonuçlar üretebildiğini ve sınırlı performans kaybı ile uygulanabilir olduğunu ortaya koymaktadır.

Wisconsin Diagnostic Breast Cancer (WDBC) veri seti ile gerçekleştirilen deneyler sonucunda, klasik makine öğrenmesi modeli (şifresiz model) ile elde edilen doğruluk oranı %97,20 olarak hesaplanmıştır. Aynı modelin homomorfik şifreleme

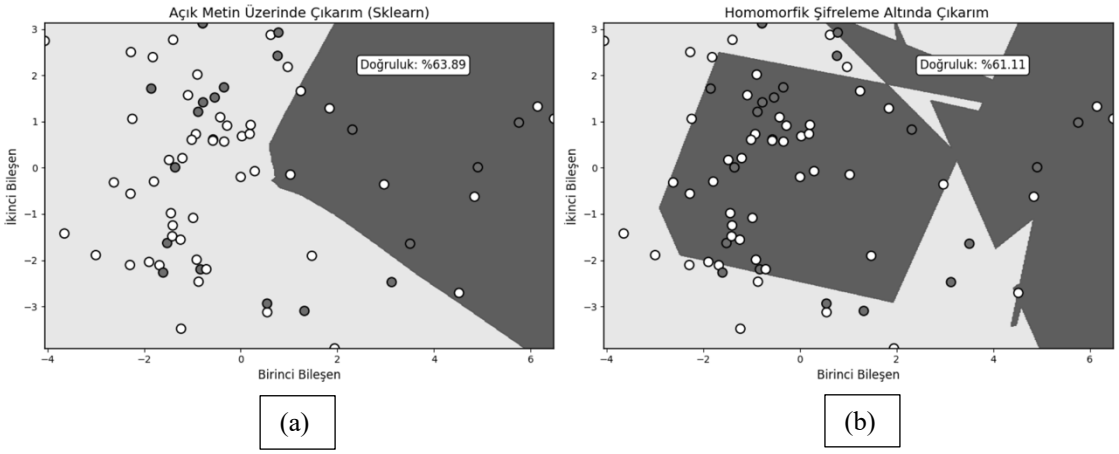
tabanlı versiyonu olan Concrete ML kullanılarak gerçekleştirilen simülasyon sonucunda ise doğruluk oranı %88,81 olarak elde edilmiştir.



Şekil 4.10. Wisconsin Diagnostic Breast Cancer veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması

Şekil 4.10 da elde edilen sonuçlar, homomorfik şifreleme kullanıldığında model performansında belirli bir düşüş yaşandığını göstermektedir. Bu durumun temel nedeni, şifreli ortamda hesaplama yapılabilmesi için uygulanan quantization (nicemleme) işlemi ve sınırlı hesaplama hassasiyetidir. Bununla birlikte, doğruluk oranındaki bu düşüşe rağmen modelin kabul edilebilir seviyede performans gösterdiği ve veri gizliliğinin korunması açısından önemli bir avantaj sağladığı görülmektedir. WDBC veri seti, sınıflar arasında yüksek ayırt edilebilirlik sunan ve genellikle yüksek doğruluk oranlarının elde edildiği bir veri setidir. Bu nedenle, şifreleme sürecinde oluşan küçük sapmalar dahi model performansında daha belirgin düşüşlere yol açabilmektedir. Başka bir ifadeyle, yüksek doğruluk gerektiren veri setlerinde homomorfik şifreleme etkileri daha hassas bir şekilde ortaya çıkmaktadır. Elde edilen bulgular, homomorfik şifreleme yöntemlerinin veri gizliliğini sağlama açısından güçlü bir yaklaşım sunduğunu, ancak yüksek hassasiyet gerektiren uygulamalarda performans kaybının daha dikkatli değerlendirilmesi gerektiğini göstermektedir. Bu durum, özellikle tıbbi karar destek sistemleri gibi kritik alanlarda FHE tabanlı modellerin kullanımında performans-gizlilik dengesi açısından önemli bir değerlendirme kriteri oluşturmaktadır.

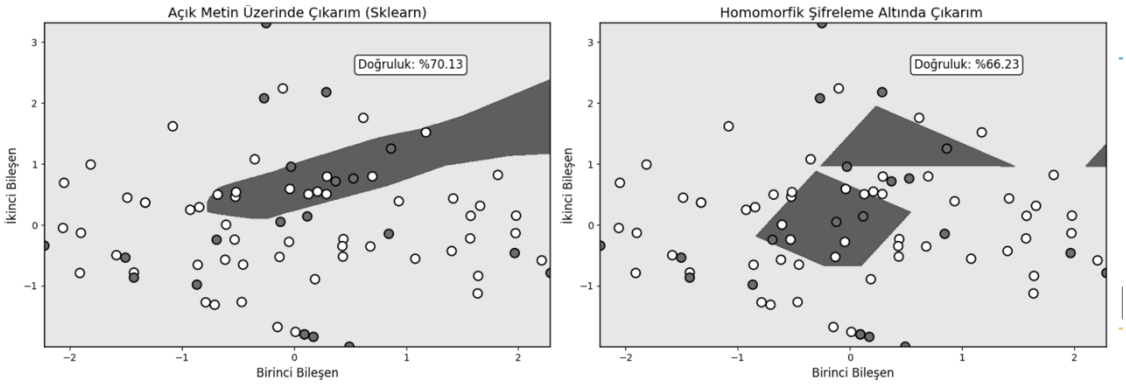
Breast Cancer veri seti meme kanseri hastalarında hastalığın tekrarlanma durumunu tahmin etmeyi amaçladığı için homomorfik şifrelenme uygulanmış ve sonuçlar bulunmuştur.



Şekil 4.11. Breast Cancer veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması

Şekil 4.11'de Breast Cancer veri seti için açık veri üzerinde eğitilen Sklearn modeli (a) ile tam homomorfik şifreleme kapsamında Concrete ML (TFHE) tabanlı modelin (b) karar sınırları karşılaştırmalı olarak sunulmuştur. Elde edilen sonuçlara göre, açık veri üzerinde çalışan model %63,89 doğruluk oranı elde ederken, homomorfik şifreleme altında çalışan model %61,11 doğruluk oranına ulaşmıştır. Bu durum, şifreleme ve nicemleme (quantization) süreçlerinin model performansında yaklaşık %2,78 oranında bir azalmaya neden olduğunu göstermektedir. Breast Cancer veri setinde sınıflar arasındaki ayırt edilebilirliğin sınırlı olması ve veri noktalarının kısmen örtüşmesi, model performansının genel olarak orta seviyede kalmasına neden olmaktadır. Bu durum, homomorfik şifreleme sürecinin etkilerinin daha görünür hale gelmesine yol açmaktadır.

Haberman's Survival veri seti, meme kanseri nedeniyle ameliyat geçiren hastaların uzun dönem hayatta kalma durumlarını incelemek amacıyla oluşturulmuş klasik bir sınıflandırma veri setine de uygulanmıştır. Şekil 4.12'de Haberman's Survival veri seti için açık veri üzerinde eğitilen Sklearn modeli ile tam homomorfik şifreleme kapsamında Concrete ML (TFHE) tabanlı modelin karar sınırları karşılaştırmalı olarak sunulmuştur.



(a) (b)

Şekil 4.12. Haberman's Survival veri seti üzerinde açık metin (a) ve homomorfik şifreleme altında gerçekleştirilen çıkarım (b) sonuçlarının karşılaştırılması

Şekil 4.12' de elde edilen sonuçlara göre, açık veri üzerinde (a) çalışan model %70,13 doğruluk oranı elde ederken, homomorfik şifreleme altında (b) çalışan model %66,23 doğruluk oranına ulaşmıştır. Bu durum, şifreleme ve nicemleme (quantization) süreçlerinin model performansında yaklaşık %3,90 oranında bir azalmaya neden olduğunu göstermektedir. Grafikler incelendiğinde, açık veri üzerinde çalışan modelin karar sınırlarının daha dengeli ve veri dağılımına kısmen uyumlu olduğu görülmektedir. Buna karşılık, homomorfik şifreleme altında çalışan modelin karar sınırlarında daha parçalı, düzensiz ve keskin geçişler oluştuğu dikkat çekmektedir. Bu farklılıklar, şifreli uzayda gerçekleştirilen hesaplamalar sırasında ortaya çıkan nicemleme hataları ve gürültü (noise) etkilerinden kaynaklanmaktadır. Haberman's Survival veri seti, sınıflar arasında belirgin bir ayrımın bulunmadığı ve veri noktalarının önemli ölçüde örtüştüğü bir yapıya sahiptir. Bu nedenle, modelin sınıflandırma başarısı genel olarak sınırlı kalmakta ve homomorfik şifreleme sürecinin etkileri daha belirgin hale gelmektedir. Elde edilen bulgular, sınıflar arasındaki ayırt edilebilirliğin düşük olduğu veri setlerinde homomorfik şifreleme kaynaklı sapmaların model performansını daha fazla etkileyebileceğini göstermektedir. Bununla birlikte, elde edilen sonuçlar FHE tabanlı modellerin bu tür zorlu veri setlerinde dahi kabul edilebilir doğruluk seviyeleri ile çalışabildiğini ve güvenli veri işleme açısından uygulanabilir olduğunu ortaya koymaktadır.

Bu çalışmada farklı yapısal özelliklere sahip toplam 12 veri seti üzerinde, klasik makine öğrenmesi modeli ile homomorfik şifreleme tabanlı Concrete ML modeli karşılaştırılmıştır. Elde edilen sonuçlar, veri setinin yapısına bağlı olarak homomorfik

şifrelemenin model performansı üzerindeki etkisinin değişkenlik gösterdiğini ortaya koymaktadır.

Öncelikle genel eğilim incelendiğinde, homomorfik şifreleme kullanımı çoğu veri setinde küçük ila orta düzeyde doğruluk kaybına neden olmuştur. Bu durum özellikle Wisconsin Diagnostic Breast Cancer (WDBC), BUPA ve Haberman veri setlerinde belirgin şekilde gözlemlenmiştir. Bu veri setlerinde doğruluk düşüşünün temel nedeni, homomorfik şifreleme sürecinde kullanılan quantization (nicemleme) işlemi ve sınırlı hesaplama hassasiyetidir. Özellikle WDBC veri setinde yüksek boyutlu ve hassas özelliklerin bulunması, nicemleme sonrası bilgi kaybını artırarak model performansını düşürmüştür.

Buna karşılık bazı veri setlerinde **(**SAHeart, Heart Disease, German Credit, UCI Breast Cancer**)** homomorfik modelin doğruluğunun klasik modele yakın olduğu veya hafif bir artış gösterdiği görülmüştür. Bu durum, özellikle düşük boyutlu veya kategorik ağırlıklı veri setlerinde, quantization sürecinin modele daha az zarar verdiğini göstermektedir. Ayrıca bu veri setlerinde modelin karar sınırlarının daha basit olması, şifreli ortamda hesaplama yapılmasını kolaylaştırmıştır.

Grafikler incelendiğinde bu durum daha açık şekilde gözlemlenmektedir. Klasik modelde karar sınırları genellikle daha düzgün ve lineer/yarı-lineer bir yapı gösterirken, homomorfik şifreleme ile elde edilen grafiklerde karar sınırlarının daha parçalı, köşeli ve düzensiz olduğu dikkat çekmektedir. Bunun nedeni, homomorfik şifreleme altında çalışan modellerin sürekli değerler yerine ayrık değerler üzerinden işlem yapmasıdır. Bu da özellikle karar sınırlarında kırılmalar ve küçük bölgeler oluşmasına yol açmaktadır.

WDBC ve Credit Approval veri setlerinde karar sınırlarının homomorfik modelde daha köşeli olduğu görülmektedir. Haberman ve BUPA gibi veri setlerinde ise bu düzensizlik doğrudan sınıflandırma hatalarına dönüşmüştür. Buna karşılık Parkinson veri setinde grafikler neredeyse aynı olup doğruluk tamamen korunmuştur; bu da veri yapısının homomorfik hesaplamaya oldukça uygun olduğunu göstermektedir.

Bir diğer önemli gözlem, veri setinin boyutu ve sınıf ayrılabilirliği ile ilgilidir. Az sayıda özellik içeren ve sınıflar arası ayrımın zayıf olduğu veri setlerinde (örneğin Haberman) klasik ve şifreli modeller düşük performans göstermiştir. Buna karşın sınıfların belirgin şekilde ayrıldığı veri setlerinde (örneğin WDBC ve Wine), klasik model çok yüksek doğruluk sağlarken, homomorfik model bu performansı tam olarak koruyamamıştır. Genel olarak elde edilen bulgular, homomorfik şifreleme kullanımının,

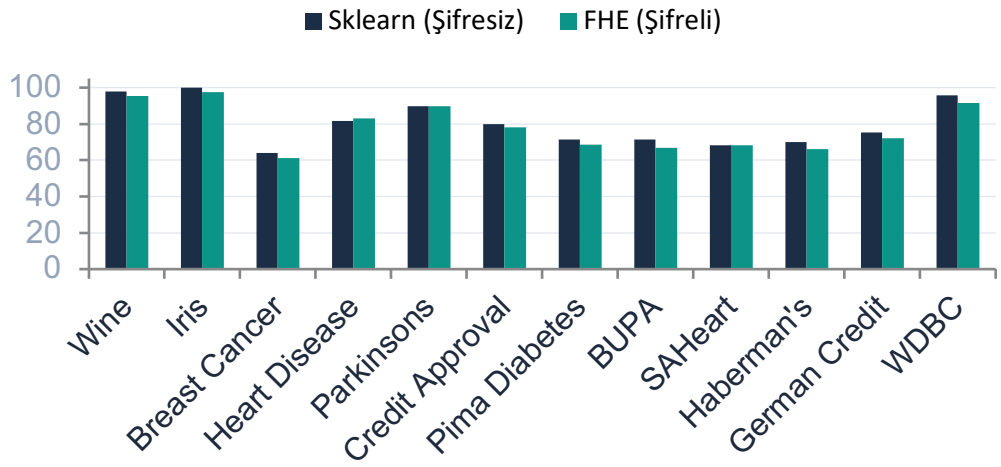
veri gizliliği açısından büyük avantaj sağladığını ancak hesaplama hassasiyeti ve model karmaşıklığı nedeniyle performans kaybına yol açabileceğini göstermektedir.

Bununla birlikte, bazı veri setlerinde performansın korunması veya artması, uygun veri yapısı ve model seçimi ile homomorfik şifreleme altında da etkili makine öğrenmesi uygulamalarının mümkün olduğunu ortaya koymaktadır. Çizelge 4.1'de şifresiz model ile tam homomorfik şifreleme yöntemlerinden torus tamamen homomorfik şifreleme yöntemi ile şifrelenmiş modelin sonuçlarının karşılaştırılması yapılmıştır.

Çizelge 4.1. Şifresiz model ve FHE model karşılaştırması

Veri seti	Örnek Sayısı	Özellik Sayısı	Açık Metin Üzerinde Çıkarım (Sklearn) Sonuçları	Homomorfik Şifreleme Altında Çıkarım Sonuçları	Sonuçlar Arasındaki Fark
Wine	178	13	97,78	95,56	-2,22
Iris	150	4	100,00	97,37	-2,63
Breast Cancer	286	9	63,89	61,11	-2,78
Heart Disease	303	51	81,58	82,89	+1,31
Parkinsons	197	22	89,80	89,80	0,00
Credit Approval	690	15	79,77	78,03	-1,74
Pima Indias Diabetes	768	8	71,35	68,75	-2,60
Bupa	345	5	71,26	66,67	-4,59
SAHeart	462	10	68,10	68,10	0,00
Haberman's Survival	306	3	70,13	66,23	-3,90
German Credit	1000	20	75,20	72,00	-3,20
WDBC	569	30	95,80	91,61	-4,19

Şekil 4.13'te farklı veri setleri üzerinde sklearn tabanlı şifresiz model ile FHE ortamında çalışan şifreli modelin doğruluk oranları karşılaştırılmıştır.



Şekil 4.13. Sklearn model (şifresiz) ile FHE model (şifreli) üzerinde veri setlerinin doğruluk oranlarının karşılaştırılması

Elde edilen sonuçlar incelendiğinde, homomorfik şifreleme uygulanmasına rağmen çoğu veri setinde doğruluk değerlerinin birbirine oldukça yakın olduğu görülmektedir. Ortalama doğruluk farkı %2,21 olarak hesaplanmıştır. En yüksek doğruluk farkı %4,59 ile BUPA veri setinde gözlemlenirken, SAHeart ve Parkinson veri setlerinde iki model arasında anlamlı bir fark oluşmamıştır. Ayrıca Heart Disease veri setinde FHE modelinin sklearn modelinden daha yüksek doğruluk elde ettiği görülmüştür. Bu sonuçlar, şifreli çıkarım işlemlerinin veri gizliliğini korurken kabul edilebilir doğruluk seviyeleri sağlayabildiğini göstermektedir.

4.3. Normalizasyon ve Ölçekleme Yöntemlerinin Etkisi

Bu çalışmada tamamen şifrelenmiş veriler üzerinde yapay sinir ağı yöntemi kullanılmıştır. Yapay sinir ağlarında giriş verilerinin uygun şekilde ölçeklendirilmesi, öğrenme sürecinin kararlılığı ve performansı için ön işleme önemli bir adımdır (Haykin, 2009) .

Çalışmada ön işleme yöntemlerinden Z-score normalizasyonu ve Min-Max ölçekleme yöntemleri tamamen homomorfik şifreleme ile uyumlu modeller açısından kritik öneme sahiptir. Homomorfik şifreleme altında çalışan makine öğrenmesi yöntemlerinden yapay sinir ağlarında, model girişlerinin uygun biçimde ölçeklenmesi ve normalizasyon işlemi yalnızca geleneksel öğrenme süreçlerinde değil aynı zamanda şifreli hesaplamaların sayısal kararlılığı ve verimliliği açısından da temel bir gerekliliktir. Literatürde de belirtildiği üzere, tamamen homomorfik şifreleme ile çalışacak veriler şifrelenmeden önce Min-Max normalizasyon veya Z-score standartlaştırma yöntemi gibi tekniklerle tutarlı aralıklara dönüştürülmelidir (Han vd., 2011).

Şifreli hesaplamalar sırasında sayısal tutarlılığı korumak ve şifreleme kaynaklı gürültü birikimini azaltmak için şarttır. Min-Max ölçekleme veri değerlerini belirli bir aralık içine sokarak hesaplamaların daha stabil yürütülmesine yardımcı olurken, Z-score ise farklı ölçeklerdeki özellikler arasındaki dengesizlikleri gidererek modelin öğrenme sürecini iyileştirir (Han vd., 2011).

Z-Score standardizasyonu uygulanan veri setlerinde sklearn tabanlı şifresiz model ile FHE ortamında çalışan şifreli modelin doğruluk sonuçları karşılaştırılmıştır. Z-Score standardizasyonunun etkileri çizelge 4.2’de gösterilmektedir.

Çizelge 4.2. Z-Score standardizasyonu uygulanan veri setlerinde sklearn ve FHE modellerinin doğruluk sonuçlarının karşılaştırılması

Veri seti	Sklearn (Şifresiz)	FHE	FHE (Normalize Edilmemiş)	Sklearn (Normalize edilmemiş)
Wine	97,78	95,56	40,00	82,22
Iris	97,37	97,37	89,47	97,37
Breast Cancer	63,89	66,67	61,11	62,50
Heart Disease	84,21	64,47	46,05	84,21
Parkinsons	89,80	89,80	32,65	81,63
Credit Approval	79,77	78,03	72,40	73,60
Pima Indias Diabetes	71,35	68,75	60,42	68,23
Bupa	71,26	66,67	64,37	67,82
SAHeart	68,10	68,10	66,38	71,55
Haberman's Survival	70,13	66,23	70,13	70,13
German Credit	75,20	72,00	58,98	78,61
WDBC	97,20	88,81	89,51	92,31

Çizelge 4.2'e göre Z-Score standardizasyonu uygulanan veri setlerinde sklearn tabanlı şifresiz model ile FHE ortamında çalışan şifreli modelin doğruluk sonuçları karşılaştırılmıştır. Elde edilen sonuçlar incelendiğinde, Z-Score standardizasyonunun veri setlerindeki farklı ölçekleri dengeleyerek model performansını olumlu yönde etkilediği gözlemlenmiştir. Şifreli çıkarım işlemleri sonucunda elde edilen doğruluk değerlerinin çoğu veri setinde sklearn modeli ile benzer sonuçlar ürettiği görülmüştür. Bazı veri setlerinde küçük doğruluk kayıpları meydana gelirken, bazı veri setlerinde ise FHE modelinin şifresiz modele yakın veya daha yüksek sonuçlar üretebildiği belirlenmiştir. Z-Score standardizasyonunun özellikle farklı varyanslara sahip özniteliklerin bulunduğu veri setlerinde daha kararlı sonuçlar sağladığı gözlemlenmiştir. Bu durum, veri ölçeklerinin dengelenmesinin hem yapay sinir ağlarının öğrenme sürecine hem de homomorfik şifreleme tabanlı çıkarım işlemlerine olumlu katkı sağladığını göstermektedir.

Min-Max normalizasyonu, veri değerlerini belirli bir aralığa dönüştürerek öznitelikler arasındaki ölçek farklılıklarını azaltmayı amaçlayan yaygın bir veri ön işleme yöntemidir (Han et al., 2011). Bu çalışmada Min-Max normalizasyonu kullanılarak veri değerleri [0,1] aralığında yeniden ölçeklendirilmiştir. Böylece veri setlerinde yer alan farklı değer aralıklarının model performansı üzerindeki etkisinin azaltılması hedeflenmiştir.

Min-Max [0,1] normalizasyonu uygulanan veri setleri üzerinde sklearn tabanlı şifresiz model ile FHE ortamında çalışan şifreli modelin doğruluk sonuçları çizelge 4.3' te verilmiştir.

Çizelge 4.3. Min-Max [0,1] normalizasyonu uygulanmış ve uygulanmamış veri setlerinde sklearn ve FHE modellerinin doğruluk sonuçları

Veri seti	Sklearn	FHE	FHE (Normalize Edilmemiş)	Sklearn (Normalize edilmemiş)
Wine	97,78	95,56	40,00	82,22
Iris	100,00	97,37	89,47	97,37
Breast Cancer	56,94	61,11	61,11	62,50
Heart Disease	78,95	82,89	46,05	84,21
Parkinsons	91,84	91,84	32,65	81,63
Credit Approval	79,19	82,66	72,40	73,60
Pima Indias Diabetes	76,56	76,04	60,42	68,23
Bupa	70,11	68,97	64,37	67,82
SAHeart	74,14	70,69	66,38	71,55
Haberman's Survival	70,13	72,73	70,13	70,13
German Credit	71,20	73,60	58,98	78,61
WDBC	94,41	95,80	89,51	92,31

Çizelge 4.3'e göre Z-Score standardizasyonu uygulanan veri setlerinde sklearn tabanlı şifresiz model ile FHE ortamında çalışan şifreli modelin doğruluk sonuçlarının birbirine yakın olduğu görülmektedir. Özellikle Iris ve Parkinsons veri setlerinde her iki model aynı doğruluk değerlerini üretmiştir. Breast Cancer veri setinde ise FHE modeli şifresiz modele göre daha yüksek doğruluk sonucu elde etmiştir.

Normalize edilmemiş veri setleri ile gerçekleştirilen şifreli çıkarım işlemlerinde bazı veri setlerinde önemli doğruluk kayıpları meydana geldiği gözlemlenmiştir. Çalışmada Min-Max normalizasyonu kullanılarak veri değerleri [-1,1] aralığında da dönüştürülmüştür. Böylece negatif ve pozitif değerlerin daha dengeli şekilde temsil edilmesi amaçlanmıştır. Min-Max [-1,1] normalizasyonu uygulanan veri setleri üzerinde sklearn tabanlı şifresiz model ile FHE ortamında çalışan şifreli modelin doğruluk sonuçları çizelge 4.4' te gösterilmiştir.

Çizelge 4.4. Min-Max [-1,1] normalizasyonu uygulanmış ve uygulanmamış veri setlerinde sklearn ve FHE modellerinin doğruluk sonuçları

Veri seti	Sklearn	FHE	FHE (Normalize Edilmemiş)	Sklearn (Normalize edilmemiş)
Wine	95,56	95,56	40,00	82,22
Iris	100,00	94,74	89,47	97,37
Breast Cancer	65,28	65,28	61,11	62,50
Heart Disease	78,95	82,89	46,05	84,21
Parkinsons	93,88	95,92	32,65	81,63
Credit Approval	81,50	82,08	72,40	73,60
Pima Indias Diabetes	70,83	72,92	60,42	68,23
Bupa	68,97	64,37	64,37	67,82
SAHeart	69,83	75,00	66,38	71,55
Haberman's Survival	70,13	66,23	70,13	70,13
German Credit	70,80	69,60	58,98	78,61
WDBC	93,71	93,71	89,51	92,31

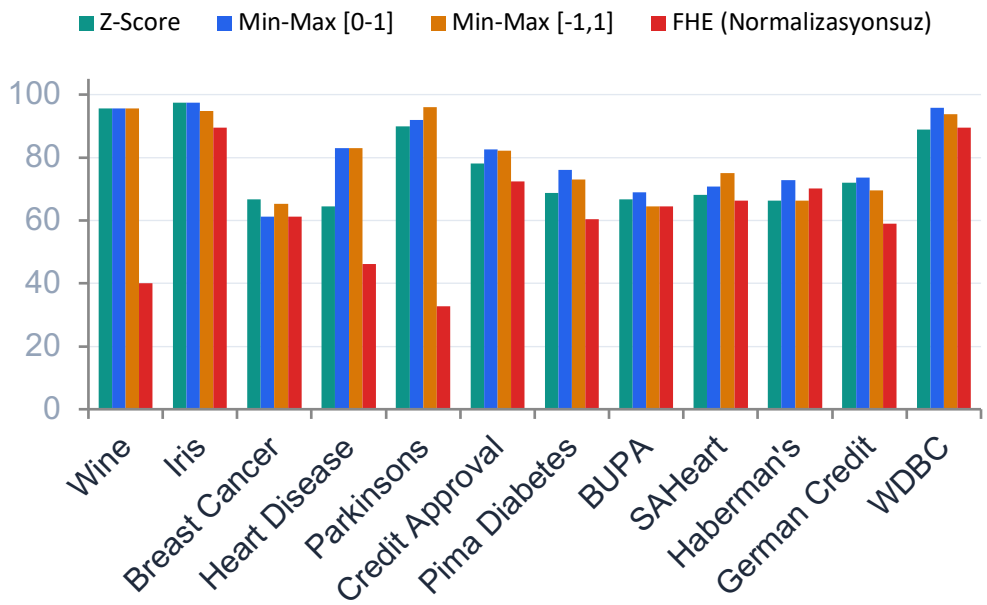
Çizelge 4.4'e göre Min-Max [-1,1] normalizasyonu uygulanan veri setlerinde sklearn tabanlı şifresiz model ile FHE ortamında çalışan şifreli modelin doğruluk sonuçları karşılaştırılmıştır. Elde edilen sonuçlar incelendiğinde, birçok veri setinde FHE modelinin sklearn modeli ile yakın doğruluk değerleri ürettiği görülmektedir. Wine, Breast Cancer ve WDBC veri setlerinde sklearn ve FHE modelleri aynı doğruluk değerlerine ulaşmıştır. Heart Disease, Parkinsons, Credit Approval, Pima Diabetes ve SAHeart veri setlerinde ise FHE modelinin sklearn modelinden daha yüksek doğruluk sonuçları verdiği görülmektedir. Normalize edilmemiş verilerle elde edilen FHE sonuçları incelendiğinde, bazı veri setlerinde belirgin doğruluk kayıpları oluştuğu görülmektedir. Özellikle Wine veri setinde doğruluk oranı %40,00'a, Parkinsons veri setinde ise %32,65'e düşmüştür. Min-Max [-1,1] normalizasyonu sonrasında bu değerlerin önemli ölçüde yükselmesi, normalizasyon işleminin model performansı üzerindeki olumlu etkisini göstermektedir.

Çizelge 4.5'te Min-Max [0,1], Min-Max [-1,1] ve Z-Score normalizasyon yöntemlerinin tamamı ve doğruluk oranları verilmiştir. Normalize edilmemiş sonuçlarla karşılaştırılmış ve en yüksek sonuçlar belirtilmiştir.

Çizelge 4.5. Normalizasyon yöntemlerine ve model türüne göre sınıflandırma doğrulukları (%)

Veri seti	MinMax (0, 1)		Z-score		MinMax (−1, 1)		Normalize edilmemiş	
	Sklearn	FHE	Sklearn	FHE	Sklearn	FHE	Sklearn	FHE
Wine	97,78	95,56	97,78	95,56	95,56	95,56	82,22	40,00
Iris	97,37	97,37	100,00	97,37	100,00	94,74	97,37	89,47
Breast Cancer	63,89	66,67	56,94	61,11	65,28	65,28	62,50	61,11
Heart Disease	84,21	64,47	78,95	82,89	78,95	82,89	84,21	46,05
Parkinsons	89,80	89,80	91,84	91,84	93,88	95,92	81,63	32,65
Credit Approval	79,77	78,03	79,19	82,66	81,50	82,08	73,60	72,40
Pima Diabetes	71,35	68,75	76,56	76,04	70,83	72,92	68,23	60,42
BUPA	71,26	66,67	70,11	68,97	68,97	64,37	67,82	64,37
SAHeart	68,10	68,10	74,14	70,69	69,83	75,00	71,55	66,38
Haberman's	70,13	66,23	70,13	72,73	70,13	66,23	70,13	70,13
German Credit	75,20	72,00	71,20	73,60	70,80	69,60	78,61	58,98
WDBC	97,20	88,81	94,41	95,80	93,71	93,71	92,31	89,51

Not: Yeşil hücreler her veri seti için en yüksek FHE doğruluğunu; turuncu hücreler normalize edilmemiş verilerle elde edilen FHE doğruluğunu göstermektedir.



Şekil 4.14. Z-Score, Min-Max [0,1], Min-Max [-1,1] ve normalizasyonsuz FHE doğruluk sonuçlarının karşılaştırılması

Şekil 4.14 incelendiğinde, farklı normalizasyon yöntemlerinin FHE model performansı üzerinde önemli etkiler oluşturduğu görülmektedir. Elde edilen sonuçlara göre Z-Score normalizasyonu, FHE modellerinde genel olarak en stabil sonuçları üretmiştir. Veri setlerinin büyük bölümünde doğruluk oranlarının daha dengeli dağıldığı ve şifreli çıkarım performansının korunduğu gözlemlenmiştir. Min-Max [0,1] normalizasyonu ise bazı veri setlerinde olumlu sonuçlar üretmesine rağmen, bazı veri setlerinde belirgin doğruluk düşüşlerine yol açmıştır. Min-Max [-1,1] normalizasyonunun ise özellikle bazı veri setlerinde daha yüksek doğruluk sonuçları sağladığı görülmüştür. Buna karşılık normalizasyon uygulanmayan FHE sonuçlarında bazı veri setlerinde ciddi performans kayıpları meydana gelmiştir. Sonuçlar genel olarak değerlendirildiğinde, veri ön işleme aşamasında uygulanan normalizasyon yöntemlerinin homomorfik şifreleme tabanlı yapay sinir ağı modellerinin performansı üzerinde doğrudan etkili olduğu görülmektedir.

4.4. Sentetik Veri Üzerinde Model Davranışı

Bu çalışmada kullanılan gerçek veri setlerine ek olarak, model davranışını daha kontrollü bir ortamda incelemek amacıyla sentetik veri üretimi gerçekleştirilmiştir.

Sentetik veri kullanımı, modelin farklı veri dağılımları altında nasıl davrandığını gözlemlemek ve karar sınırlarının oluşumunu daha açık bir şekilde analiz edebilmek açısından önemli bir avantaj sağlamaktadır. Üretilen sentetik veriler 100 örnek sayısından başlayıp 100 artış ile 1000 örnek sayısında oluşturulmuştur. Üretilen sentetik veri setleri üzerinde farklı örnek sayıları ve özellik boyutları dikkate alınarak model performansı değerlendirilmiştir. Bu kapsamda elde edilen sonuçlar Çizelge 4.6'da sunulmaktadır.

Çizelge 4.6. Sentetik veri üzerinde model mimarisi ve performans sonuçları

Özellik Sayısı	Örnek Sayısı	Nöron Sayısı	Katman Sayısı	Optimum Skor	Doğruluk (%)	Kesinlik (%)	Kayıp
2	100	2	3	60	100,00	96,00	0,15
2	300	5	4	120	96,00	96,00	0,43
2	500	5	4	120	96,60	96,00	0,63
2	700	4	4	120	98,29	92,00	0,59
3	100	2	2	60	90,00	91,00	0,10

En iyi sonuçlar eklenmiştir.

Çizelge 4.6 incelendiğinde, sentetik veri üzerinde modelin genel olarak yüksek doğruluk değerlerine ulaştığı görülmektedir. Özellikle düşük özellik ve örnek sayısına sahip veri setlerinde modelin oldukça başarılı sonuçlar verdiği dikkat çekmektedir. Örneğin, 2 özellik ve 100 örnekten oluşan veri setinde model %100 doğruluk değerine ulaşmıştır. Bu durum, düşük boyutlu veri yapılarında sınıflar arasındaki ayrımın daha kolay yapılabildiğini göstermektedir. Model mimarisi açısından değerlendirildiğinde, katman sayısı ve gizli katmanlardaki nöron sayısının performans üzerinde belirleyici bir etkisi olduğu gözlemlenmiştir. Özellikle daha basit veri yapılarında daha az katman ve daha az nöron ile yüksek doğruluk değerlerine ulaşılabildiği görülmektedir. Bu durum, model karmaşıklığının veri karmaşıklığı ile uyumlu olması gerektiğini ortaya koymaktadır. Örnek sayısının artmasıyla birlikte model performansının genel olarak korunduğu, ancak küçük dalgalanmalar meydana geldiği gözlemlenmiştir. 300 ve 500 örnekli veri setlerinde doğruluk değerleri %96 civarında seyrederken, 700 örnekli veri setinde doğruluk %98,29 seviyesine ulaşmıştır. Bu durum, veri miktarının artmasının modeli desteklediğini, ancak performans artışının her zaman doğrusal olmadığını göstermektedir.

Özellik sayısının artması ile model performansında düşüş gözlemlenmiştir. 3 özellikli veri setinde doğruluk değerinin %90 seviyesine gerilemesi, veri boyutunun artmasıyla birlikte modelin karar sınırlarını belirlemesinin zorlaştığını göstermektedir. Bu durumda modelin daha karmaşık bir mimariye ihtiyaç duyabileceği değerlendirilmektedir.

Katman sayısı ve nöron sayısı birlikte değerlendirildiğinde, daha yüksek katman sayısının her zaman performans artışı sağlamadığı görülmektedir. Aksine, bazı durumlarda daha fazla katman kullanımı modelin aşırı öğrenmesine (overfitting) veya öğrenme sürecinin zorlaşmasına neden olabilmektedir. Benzer şekilde, nöron sayısının artırılması da her zaman doğruluk artışı ile sonuçlanmamaktadır. Bu bulgular, model mimarisinin veri yapısına uygun şekilde optimize edilmesi gerektiğini göstermektedir.

Kayıp değerleri incelendiğinde, örnek sayısının artmasıyla birlikte kayıp değerlerinde genel bir artış eğilimi gözlemlenmiştir. Bu durum, veri dağılımının karmaşıklığının artmasıyla modelin öğrenme sürecinde daha fazla hata ile karşılaşabileceğini göstermektedir.

Genel olarak değerlendirildiğinde, sentetik veri üzerinde yapılan analizler, model performansının yalnızca veri miktarına değil, aynı zamanda model mimarisi parametrelerine de bağlı olduğunu ortaya koymaktadır. Bu sonuçlar, katman sayısı ve nöron sayısının veri özelliklerine göre dikkatli bir şekilde seçilmesi gerektiğini göstermekte ve model tasarımının önemini vurgulamaktadır.

4.5. Hesaplama Maliyeti Analizi

Tam homomorfik şifreleme kullanılarak gerçekleştirilen hesaplamalar, veri gizliliği ve güvenliği açısından önemli avantajlar sunmasına rağmen, klasik şifresiz hesaplama yöntemleriyle karşılaştırıldığında en belirgin dezavantaj olarak yüksek hesaplama maliyeti ve uzun yürütme süreleri ile öne çıkmaktadır. FHE tabanlı sistemlerde, verilerin şifreli biçimde işlenmesi karmaşık matematiksel işlemler, büyük anahtar boyutları ve ek şifreleme/deşifreleme adımları gerektirdiğinden, işlem sürelerinde ciddi artışlara neden olmaktadır. Özellikle makine öğrenmesi ve yapay sinir ağı gibi çok sayıda aritmetik işlem içeren yöntemlerde bu durum daha da belirgin hâle gelmektedir. Bu nedenle, FHE ile gerçekleştirilen çıkarım ve model çalıştırma süreçleri, klasik şifresiz modellere kıyasla daha fazla hesaplama kaynağı tüketmekte ve yürütme süreleri açısından dezavantaj oluşturmaktadır.

Bu bölümde, şifresiz ve FHE tabanlı modellerin hesaplama maliyetleri karşılaştırmalı olarak analiz edilerek, veri seti büyüklüğü, özellik sayısı ve kullanılan ölçekleme yöntemlerinin yürütme süresine olan etkileri ayrıntılı biçimde incelenmektedir. Çalışmada homomorfik şifreleme tabanlı yapay sinir ağı modellerinin hesaplama maliyeti üç temel bileşen üzerinden incelenmiştir. Bunlar bit genişliği**,** anahtar üretme süresi ve homomorfik şifreleme ile şifrelenmiş verilerin üzerinde çalışma süresidir. Bit genişliği, modelin hesaplama sırasında işlediği sayısal değerlerin temsil kapasitesini belirlemektedir. Homomorfik şifreleme devrelerinde bit genişliği arttıkça, işlenen sayısal aralık genişlemekte ve buna bağlı olarak gerçekleştirilen işlemlerin karmaşıklığı artmaktadır. Bu durum, doğrudan hesaplama süresine yansımakta ve özellikle yüksek bit genişliğine sahip devrelerde işlem maliyetinin belirgin şekilde yükselmesine neden olmaktadır. Yüksek bit genişliğine sahip devrelerde şifreli hesaplama süresi önemli ölçüde arttığı gözlemlenmiştir.

Anahtar üretme süresi, homomorfik şifreleme için gerekli kriptografik anahtarların oluşturulması sırasında ortaya çıkan başlangıç maliyetini ifade etmektedir. Bu süreç, model çalıştırılmadan önce gerçekleştirilmekte olup genellikle tek seferlik bir maliyet olarak değerlendirilmektedir. Ancak elde edilen sonuçlar, anahtar üretme süresinin veri setine ve devre yapısına bağlı olarak değişkenlik gösterebildiğini ortaya koymaktadır. Tam homomorfik şifreleme yöntemlerinden Torus ile şifrelenmiş model üzerinde çalışma süresi, modelin şifreli veri üzerinde tahmin üretmesi için geçen süreyi ifade etmektedir. Bu süre, homomorfik işlemler, aritmetik hesaplamalar ve gerektiğinde bootstrapping işlemlerini içermektedir. Bu nedenle, sistemin toplam çalışma maliyetinin en önemli bileşeni olarak değerlendirilmektedir.

Çizelge 4.7. Hesaplama maliyet karşılaştırması

Veri seti	Örnekler	Özellikler	Veri Tipi	Anahtar Üretme Süresi (sn)	Homomorfik Şifreli Veri Üzerinde Çalışma Süresi (sn)	Bit Sayısı
Wine	178	13	Sürekli	1,08	0,66	9
Iris	150	4	Sürekli	0,45	0,37	7
Breast Cancer	286	9	Kategorik	0,67	0,51	8
Heart Disease	303	51	Karma	1,00	1,36	8
Parkinsons	197	22	Sürekli	1,13	6,93	11
Credit Approval	690	15	Karma	1,40	5,41	10
Pima Indias Diabetes	768	8	Sürekli	4,16	5,00	10
Bupa	345	5	Sürekli	0,59	0,37	8
SAHeart	462	10	Karma	3,22	1,59	8
Haberman’s Survival	306	3	Tamsayı	1,40	1,01	9
German Credit	1000	20	Karma	0,64	1,38	9
WDBC	569	30	Sürekli	1,85	5,82	10

Çizelge 4.7'de veriler incelendiğinde, yapılan çalışmada elde edilen sonuçlar homomorfik şifreleme altında gerçekleştirilen hesaplamaların maliyetinin veri setine ve model yapısına bağlı olarak önemli ölçüde değiştiğini göstermektedir. Özellikle örnek başına çıkarım süreleri incelendiğinde, en düşük değerleri İris ve SAHeart veri setlerinde yaklaşık 0,37 saniye olduğu, buna karşılık en yüksek değerin Credit Approval veri setinde 6,93 saniye olarak gerçekleştiği görülmüştür. Benzer şekilde, WDBC, Breast Cancer ve German Credit veri setlerinde de yüksek çıkarım süreleri elde edilmiştir. Tüm veri setleri birlikte değerlendirildiğinde, örnek başına ortalama çıkarım süresi yaklaşık 2,53 saniye olarak hesaplanmıştır. Bununla birlikte, medyan değerin daha düşük olması, bazı veri setlerinde maliyetin belirgin şekilde arttığını göstermektedir. Bu durum, homomorfik şifreleme sistemlerinde veri setine bağlı değişkenliğin önemli bir faktör olduğunu ortaya koymaktadır.

Bit genişliği ile hesaplama süresi arasındaki ilişki incelendiğinde, düşük bit genişliğine sahip devrelerde işlem sürelerinin daha düşük olduğu, yüksek bit genişliğine sahip devrelerde ise sürelerin önemli ölçüde arttığı görülmüştür. Özellikle 7–9 bit aralığında ortalama çıkarım süresi yaklaşık 0,91 saniye iken, 10–11 bit aralığında bu sürenin yaklaşık 5,79 saniye'ye yükseldiği belirlenmiştir. Bu durum, bit genişliğinin hesaplama maliyeti üzerinde doğrudan etkili olduğunu göstermektedir.

Anahtar üretme süreleri incelendiğinde, ortalama sürenin yaklaşık 1,65 saniye olduğu görülmektedir. En düşük anahtar üretme süresi 0,59 saniye ile İris veri setinde, en yüksek süre ise 4,16 saniye ile German Credit veri setinde elde edilmiştir. Ancak anahtar üretme süresinin tek seferlik bir maliyet olması nedeniyle, sistemin toplam çalışma süresi üzerinde sınırlı bir etkiye sahip olduğu değerlendirilmektedir.

Bu çalışmada gerçekleştirilen deneyler, Apple M4 işlemcili, 16 GB RAM ve 512 GB depolama kapasitesine sahip bir bilgisayar ortamında yürütülmüştür. Sistem, 10 çekirdekli CPU ve 10 çekirdekli GPU mimarisine sahiptir. Deneysel uygulamalar Python programlama dili kullanılarak PyCharm geliştirme ortamında gerçekleştirilmiştir. Elde edilen hesaplama süreleri, kullanılan donanım özelliklerine bağlı olarak değişkenlik gösterebilir. Sonuç olarak, bu çalışmada homomorfik şifreleme tabanlı makine öğrenmesi uygulamalarında hesaplama maliyetinin en önemli belirleyicisinin şifreli model üzerinde çalışma süresi olduğu görülmüştür. Bununla birlikte, bit genişliği maliyeti doğrudan etkileyen kritik bir parametre olarak öne çıkmaktadır. Bu nedenle, model tasarımı

sürecinde doğruluk ile bit genişliği ve hesaplama maliyeti arasındaki dengenin dikkate alınması gerekmektedir.

5. SONUÇLAR VE ÖNERİLER

Bu çalışmada, tam homomorfik şifreleme yöntemlerinden torus kullanılarak şifreli veriler üzerinde yapay sinir ağı modellerinin performansı incelenmiş ve şifresiz modellerle karşılaştırmalar yapılmıştır. Bu çalışma kapsamında Torus Homomorfik Şifreleme (TFHE) mimarisi kullanılarak geliştirilen yapay sinir ağı modeli, 12 farklı veri seti üzerinde test edilmiştir. Elde edilen bulgular, şifreli uzayda gerçekleştirilen çıkarımların, standart Sklearn modellerine kıyasla ortalama %2-4 bandında bir doğruluk kaybıyla çalıştığını, buna karşılık uçtan uca veri gizliliği sağladığını göstermektedir. Hesaplama maliyetleri açısından TFHE'nin düşük bit derinliklerinde (7-11 bit) saniyeler mertebesinde sonuç vermesi, gerçek zamanlı olmasa da gizlilik öncelikli finansal (German Credit) ve medikal (Breast Cancer) analizler için modelin yüksek potansiyele sahip olduğunu ortaya koymaktadır. Elde edilen sonuçlar ise tam homomorfik şifreleme tabanlı modellerin doğruluk açısından şifresiz modellere kıyasla önemli bir kayıp yaşamadığını göstermektedir. Bu bulgu, homomorfik şifrelemenin yalnızca model performansını korumakla kalmayıp, aynı zamanda veri gizliliğini güçlü bir şekilde sağladığını ortaya koymaktadır. FHE sayesinde, hassas veriler şifreli olarak işlenebilir ve hiçbir noktada açık hâlde sunulmadan çıkarım yapılabilir; bu durum, özellikle sağlık, finans ve kişisel veri içeren uygulamalarda kullanıcı gizliliğinin korunması açısından kritik bir avantaj sağlamaktadır. Dolayısıyla, FHE hem veri güvenliği hem de güvenilir model performansı açısından etkili ve uygulanabilir bir çözüm sunmaktadır.

5.1. Sonuçlar

Çalışma boyunca elde edilen bilgiler doğrultusunda tamamen homomorfik şifreleme (FHE) kullanılarak şifreli veriler üzerinde yapay sinir ağı modellerinin performansı değerlendirilmiş, homomorfik şifreleme tabanlı makine öğrenmesi yaklaşımlarının şifrelenmiş veriler üzerinde güvenli çıkarım yapabilme kapasitesi, farklı veri setleri ve veri ön işleme yöntemleri kullanılarak incelenmiştir. Çalışmanın temel amacı, açık veri üzerinde çalışan yapay sinir ağı ile tam homomorfik şifreleme yöntemlerinde torus ile modellerin performanslarını karşılaştırmak ve bu süreçte ortaya çıkan doğruluk kaybı ile hesaplama maliyeti arasındaki ilişkiyi değerlendirmektir.

Çalışmanın sonunda elde edilen sonuçlar, homomorfik şifreleme yöntemlerinin veri gizliliğini koruyarak yapay sinir ağlarının çalıştırılmasına olanak sağladığını açık biçimde göstermektedir. Açık veri ile karşılaştırıldığında, şifrelenmiş veriler altında

çalışan modellerin çoğu veri setinde kabul edilebilir düzeyde doğruluk sergilediği görülmüştür. Özellikle İris ve Wine veri setlerinde tam homomorfik şifreleme sonuçlarının açık veri sonuçlarıyla örtüşmesi, bu yöntemlerin uygun koşullarda performans kaybı oluşturmadan uygulanabildiğini ortaya koymaktadır. Bu durum, veri yapısının daha karmaşık olduğu veya sınıflar arası ayrımın zor olduğu veri setlerinde doğruluk farklarının daha belirgin hale geldiği gözlemlenmiştir. Özellikle German Credit ve WDBC veri setlerinde tam homomorfik şifreleme altında performans kaybı daha belirgin olmuştur. Bununla birlikte, bu veri setlerinde dahi modellerin işlevselliğini koruduğu görülmüştür. Yapılan çalışmada dikkat çeken bir diğer bulgu ise bazı veri setlerinde doğruluk oranları benzer olmasına rağmen karar sınırlarının farklılaşmasıdır. Bu durum, model performansının yalnızca doğruluk oranı üzerinden değerlendirilmesinin yeterli olmadığını, modelin karar mekanizmasının da incelenmesi gerektiğini ortaya koymaktadır.

Veri ön işleme yöntemlerinin tam homomorfik şifreleme tabanlı modeller üzerindeki etkisi de önemli bir sonuç olarak ortaya çıkmıştır. Z-score ve Min-Max normalizasyon yöntemlerinin farklı veri setlerinde farklı performanslar sergilediği görülmüş ve uygun normalizasyon yönteminin seçiminin model başarısını doğrudan etkilediği belirlenmiştir.

Hesaplama kısmı şifrelenmiş veriler ile çalışmanın en maliyetli kısmı olarak karşımıza çıkmaktadır. Hesaplama maliyet analizi, homomorfik şifreleme tabanlı sistemlerin en önemli sınırlayıcı faktörlerinden birinin işlem süresi olduğunu göstermiştir. Özellikle şifreli model üzerinde çalışma süresinin, toplam maliyetin en belirleyici bileşeni olduğu görülmüştür. Bazı veri setlerinde bu sürenin saniyenin altında kaldığı, bazı veri setlerinde ise birkaç saniyeye hatta daha yüksek değerlere ulaştığı belirlenmiştir. Bunun yanı sıra bit genişliği ile hesaplama maliyeti arasında güçlü bir ilişki olduğu gözlemlenmiştir. Düşük bit genişliğine (7-9 bit) sahip devrelerde işlem süreleri daha düşük kalırken, 10 bit ve üzerindeki devrelerde hesaplama maliyetinin belirgin şekilde arttığı tespit edilmiştir. Bu durum, bit genişliğinin homomorfik şifreleme sistemlerinde kritik bir parametre olduğunu göstermektedir. Maliyet açısından son kısım ise anahtar üretme süresi ise sistemin başlangıç maliyetini temsil etmekte olup genellikle tek seferlik bir işlem olması nedeniyle toplam maliyet üzerindeki etkisinin sınırlı olduğu sonucuna ulaşılmıştır.

Sonuç olarak, homomorfik şifreleme tabanlı makine öğrenmesi yöntemlerinden yapay sinir ağlarının veri gizliliğini koruyarak uygulanabilir olduğu, ancak bu yaklaşımın doğruluk ve hesaplama maliyeti arasında dikkatli bir denge gerektirdiği belirlenmiştir.

5.2. Öneriler

Çalışmanın bulguları doğrultusunda, tam homomorfik şifreleme tabanlı makine öğrenmesi sistemlerinden yapay sinir ağlarının geliştirilmesine yönelik çeşitli öneriler sunulmaktadır. Öncelikle, model tasarım sürecinde yalnızca doğruluk değil, aynı zamanda bit genişliği ve hesaplama maliyeti de dikkate alınmalıdır. Daha düşük bit genişliğine sahip modellerin tercih edilmesi, işlem sürelerini önemli ölçüde azaltabilir ve sistemin daha verimli çalışmasını sağlayabilir.

Veri ön işleme aşamasının model performansı üzerindeki etkisi göz önünde bulundurularak, farklı normalizasyon yöntemlerinin sistematik olarak değerlendirilmesi önerilmektedir. Özellikle Min-Max ve Z-score yöntemlerinin veri setine göre optimize edilmesi, daha başarılı sonuçlar elde edilmesine katkı sağlayacaktır.

Gelecek çalışmalarda, farklı homomorfik şifreleme şemalarının karşılaştırılması ve farklı model türlerinin (örneğin derin öğrenme modelleri) şifreli veri üzerinde performanslarının incelenmesi önerilmektedir.

Hesaplama maliyetinin azaltılması amacıyla, daha verimli nicemleme yöntemlerinin geliştirilmesi ve devre optimizasyonlarının yani hesaplamayı daha hızlı ve daha düşük maliyetle yapılması önerilmektedir. Bunun yanında donanımsal hızlandırıcıların kullanımı da performans açısından önemli kazanımlar sağlayabilir.

Son olarak, homomorfik şifreleme tabanlı yöntemlerin sağlık, finans ve kamu gibi veri gizliliğinin kritik olduğu alanlarda uygulanabilirliğinin artırılması için çalışmaların genişletilmesi önerilmektedir.

6. KAYNAKLAR

Acar, A., Aksu, H., Uluagac, A. S. ve Conti, M. (2019). A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys, 51(4). https://doi.org/10.1145/3214303

Aggarwal, C. C. (2018). Neural networks and deep learning. Springer. https://doi.org/10.1007/978-3-319-94463-0

Khan, A. N., Fan, M. Y., Malik, A. Ve Memon, R.A. (2019). Learning from privacy preserved encrypted data on cloud through supervised and unsupervised machine learning. In 2019 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET). IEEE.

Altıntaş, S. ve Barkuş, M. (2023). Dijital ortamlarda kişisel veri güvenliği kavramı üzerine bir derleme çalışması. Electronic Journal of Vocational Colleges, 13, 46-69.

Kâhya, A. (2022). Machine learning over encrypted data with fully homomorphic encryption [Yüksek lisans tezi, Middle East Technical University].

Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

Cheon, J. H., Kim, A., Kim, M. ve Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. Lecture Notes in Computer Science, 10624, 409–437. https://doi.org/10.1007/978-3-319-70694-8_15

Chillotti, I., Gama, N., Georgieva, M. ve Izabachène, M. (2016). TFHE: Fast fully homomorphic encryption over the torus. Cryptology ePrint Archive, Report 2018/421. https://eprint.iacr.org/2018/421

Chillotti, I. (2022, Mayıs 4). *TFHE deep dive – Part I: Ciphertext types*. Zama. https://www.zama.org/post/tfhe-deep-dive-part-1 [Erişim tarihi: 19.04.2026].

Chillotti, I. (2022, Mayıs 11). *TFHE deep dive – Part II: Encodings and linear leveled operations*. Zama. https://www.zama.org/post/tfhe-deep-dive-part-2 [Erişim tarihi: 19.04.2026].

Chillotti, I. (2022, Mayıs 18). *TFHE deep dive – Part III: Key switching and leveled multiplication*. Zama. https://www.zama.org/post/tfhe-deep-dive-part-3 [Erişim tarihi: 19.04.2026].

Chillotti, I. (2022, Haziran 2). *TFHE deep dive – Part IV: Programmable bootstrapping*. Zama. https://www.zama.org/post/tfhe-deep-dive-part-4 [Erişim tarihi: 19.04.2026].

Chaudhary, P., Gupta, R., Singh, A. ve Majumder, P. (2019). Analysis and comparison of various fully homomorphic encryption techniques. 2019 International Conference on Computing, Power and Communication Technologies (GUCON). (s.58-62)IEEE.

Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. STOC '09: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing (s. 169–178). ACM. https://doi.org/10.1145/1536414.1536440

Çetin, F. (2021). Kriptografide kullanılan asal sayı test yöntemleri üzerine bir çalışma. [Yüksek lisans tezi, Necmettin Erbakan Üniversitesi, Fen Bilimleri Enstitüsü]

Dua, D. ve Graff, C. (2019a). Breast Cancer Wisconsin (Original) Data Set. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original)

Dua, D. ve Graff, C. (2019b). Liver Disorders (BUPA) Data Set. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/liver+disorders

Dua, D. ve Graff, C. (2019c). Credit Approval Data Set. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/credit+approval

Dua, D. ve Graff, C. (2019d). Statlog (German Credit Data) Data Set. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

Dua, D. ve Graff, C. (2019e). Haberman's Survival Data Set. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/haberman

Dua, D. ve Graff, C. (2019f). Heart Disease Data Set. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/heart+disease

Dua, D. ve Graff, C. (2019g). Iris Data Set. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/iris

Dua, D. ve Graff, C. (2019h). Parkinsons Data Set. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/parkinsons

Dua, D. ve Graff, C. (2019i). Pima Indians Diabetes Data Set. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes

Dua, D. ve Graff, C. (2019j). Breast Cancer Wisconsin (Diagnostic) Data Set. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)

Dua, D. ve Graff, C. (2019k). Wine Data Set. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/wine

Goodfellow, I., Bengio, Y. ve Courville, A. (2016). Deep learning. MIT Press. https://www.deeplearningbook.org

Hamza, R. (2023). Homomorphic encryption for AI-based applications: Challenges and opportunities. Proceedings – International Conference on Knowledge and Systems Engineering, KSE. https://doi.org/10.1109/KSE59128.2023.10299436

Han, J., Kamber, M. ve Pei, J. (2011). Data mining: Concepts and techniques (3. baskı). Morgan Kaufmann.

Hastie, T., Tibshirani, R. ve Friedman, J. (2009). South African Heart Disease Data Set. https://hastie.su.domains/ElemStatLearn/datasets/SAheart.data

Haykin, S. S. (2009). Neural networks and learning machines (3. baskı). Prentice Hall/Pearson.

Hikmat Mahmood, Z. ve Khalel Ibrahem, M. (2018). New fully homomorphic encryption scheme based on multistage partial homomorphic encryption applied in cloud computing. In 2018 1st Annual International Conference on Information and Sciences [AICIS] (pp. 182-187). IEEE. https://doi.org/10.1109/AİCIS.2018.00043

Hoşcoşkun, R. E. (2020). Homomorfik şifreleme yöntemi üzerine bir inceleme [Yayınlanmamış yüksek lisans tezi]. Trakya Üniversitesi Fen Bilimleri Enstitüsü.

Kabasakaloğlu, M. U. ve Eyüpoğlu, C. (2025). Nesnelerin internetinde veri mahremiyetinin korunması üzerine bir inceleme. İstanbul Ticaret Üniversitesi Teknoloji ve Uygulamalı Bilimler Dergisi, 8(1), 143–150. https://doi.org/10.56809/icujtas.1630096

Kadykov, V., Levina, A. ve Grebenevich, K. (2024). Implementation of lattices system in homomorphic encryption. International Conference on Artificial Intelligence, Computer, Data Sciences, and Applications (ACDSA 2024). https://doi.org/10.1109/ACDSA59508.2024.10467404

Kim, W. ve Seok, J. (2022). Privacy-preserving collaborative machine learning in biomedical applications. 4th International Conference on Artificial Intelligence in Information and Communication (ICAIIC 2022) Proceedings (s. 179–183). https://doi.org/10.1109/ICAIIC54071.2022.9722703

Kogos, K. G., Filippova, K. S. ve Epishkina, A. V. (2017). Fully homomorphic encryption schemes: The state of the art. In 2017 International Conference on Computing, Power and Communication Technologies (s.463-466). IEEE.

Kushilevitz, E. ve Malkin, T. (Ed.). (2016). Lecture notes in computer science: C. 9562. Theory of cryptography. Springer. https://doi.org/10.1007/978-3-662-49096-9

LeCun, Y., Bengio, Y. ve Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, C. H., Lim, K. H. ve Eswaran, S. (2025). A comprehensive survey on secure healthcare data processing with homomorphic encryption: Attacks and defenses. Discover Public Health, 22(1). https://doi.org/10.1186/s12982-025-00505-w

Li, W. (2026). Theoretical framework and application exploration of fully homomorphic encryption. Highlights in Science, Engineering and Technology SDPIT.

Liu, W., You, L., Shao, Y., Shen, X., Hu, G., Shi, J. ve Gao, S. (2025). From accuracy to approximation: A survey on approximate homomorphic encryption and its applications. Computer Science Review, 55. https://doi.org/10.1016/j.cosrev.2024.100689

McCulloch, W. S. ve Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5(4), 115–133. https://doi.org/10.1007/BF02478259

Mert, A. C., Ozturk, E. ve Savas, E. (2020). Design and implementation of encryption/decryption architectures for BFV homomorphic encryption scheme. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 28(2), 353–362. https://doi.org/10.1109/TVLSI.2019.2943127

Müller, A. C. ve Guido, S. (2016). Introduction to machine learning with Python: A guide for data scientists. O'Reilly Media.

Öztürk, K. (2018). Yapay sinir ağları ve yapay zekâya genel bir bakış. TAKVİM Akademik Sosyal Bilimler Dergisi, 6(2), 25–36. https://dergipark.org.tr/tr/pub/takvim/article/427526

Regev, O. (2009). On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM, 56(6), 1–40. https://doi.org/10.1145/1568318.1568324

Rumelhart, D. E., Hinton, G. E. ve Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536. https://doi.org/10.1038/323533a0

Sağıroğlu, Ş. ve Akleylek, M. (2021). Siber güvenlik ve savunma: Blokzincir ve kriptoloji – 5. Nobel Akademik Yayıncılık.

Srinivasa Rao, B., Chattopadhyay, S., Singh, P., Hazela, B., Sabarinathan, G. ve Yamini, K. (2023). Privacy-aware artificial intelligence with homomorphic encryption using machine learning. International Conference on Sustainable Computing and Smart Systems (ICSCSS 2023) Proceedings (s. 259–265). https://doi.org/10.1109/ICSCSS57650.2023.10169776

Stallings, W. (2009). Cryptography and network security: Principles and practice. Pearson/Prentice Hall.

T'Jonck, K., Kancharla, C. R., Pang, B., Hallez, H. ve Boydens, J. (2022). Privacy preserving classification via machine learning model inference on homomorphic encrypted medical data. 2022 31st International Scientific Conference Electronics (ET 2022) Proceedings. https://doi.org/10.1109/ET55967.2022.9920289

Turan, F., Roy, S. S. ve Verbauwhede, I. (2020). HEAWS: An accelerator for homomorphic encryption on the Amazon AWS FPGA. IEEE Transactions on Computers, 69(8), 1185–1196. https://doi.org/10.1109/TC.2020.2988765

Wang, C., Chen, J., Zhang, X. ve Cheng, H. (2023). An efficient fully homomorphic encryption sorting algorithm using addition over TFHE. Proceedings of the International Conference on Parallel and Distributed Systems – ICPADS, 226–233. https://doi.org/10.1109/ICPADS56603.2022.00037

Yılmaz, A. ve Şimşek, C. (2026). Yapay sinir ağlarında öğrenme dinamiğini ve başarımı artırmak için standart aktivasyonların hibritleştirilmesi. Karadeniz Fen Bilimleri Dergisi, 16(1), 497–513. https://doi.org/10.31466/kfbd.1826121

Zama. (2023). Zama – Fully homomorphic encryption for machine learning. https://docs.zama.ai/concrete-ml/

EKLER

EK-1TFHE Tabanlı Homomorfik Şifreleme için Örnek Uygulama

Bu bölümde, homomorfik şifreleme sürecinin temel mantığını göstermek amacıyla hazırlanmış sadeleştirilmiş bir uygulama sunulmuştur. Bu kod, Learning With Errors (LWE) tabanlı şifreleme, homomorfik toplama, skaler çarpma ve sadeleştirilmiş programmable bootstrapping işlemlerini içermektedir. Kod yöntemin temel çalışma prensiplerini kavramsal düzeyde açıklamak amacıylageliştirilmiştir. Bu yönüyle, çalışmada kullanılan Concrete-ML altyapısının arka planını destekleyici niteliktedir.

1. Kütüphane İçe Aktarımları

Uygulamada yalnızca NumPy kütüphanesi kullanılmıştır. Veri yapıları Python'un standart dataclass modülü ile tanımlanmıştır.

```
import numpy as np
from dataclasses import dataclass
from typing import List, Tuple, Union
```

2. LWE Parametreleri

LWEParams sınıfı şifreleme sisteminin temel parametrelerini tanımlamaktadır. Gizli anahtar boyutu n = 630, modülüs q = 2^{32} ve hata standart sapması σ = 2^{-15} olarak ayarlanmıştır. Δ (delta) parametresi, düz metin mesajlarını torus uzayına ölçeklendirmek için kullanılır.

```
# PARAMETRELER
@dataclass
class LWEParams:
    n: int = 630                  # Gizli anahtar boyutu
    q: int = 2**32                # Modülüs
    std_dev: float = 2**-15       # Hata standart sapması
    plaintext_bits: int = 8       # Düz metin bit derinliği

    @property
    def delta(self) -> int:
        return self.q // (2 ** self.plaintext_bits)
```

3. Anahtar Üretimi

keygen() fonksiyonu $\{0,1\}^n$ uzayından düzgün dağılımlı rastgele bir ikili vektör seçerek gizli anahtarı oluşturur. Gizli anahtar yalnızca istemci tarafında saklanır.

```
# ANAHTAR ÜRETİMİ
@dataclass
class LWESecretKey:
    s: np.ndarray
    params: LWEParams

def keygen(params: LWEParams):
    rng = np.random.default_rng()
    s = rng.integers(0, 2, size=params.n, dtype=np.int64)
    return LWESecretKey(s=s, params=params)
```

4. Şifreli Metin Veri Yapısı

LWE şifreli metin (a, b) çifti olarak temsil edilmektedir. _mod_q() yardımcı fonksiyonu q modülüsü altında tam sayı ve dizi işlemlerini yönetir.

```
# ŞİFRELİ METİN VERİ YAPISI
@dataclass
class LWECiphertext:
    a: np.ndarray
    b: int
    params: LWEParams

def _mod_q(x, q):
    if isinstance(x, (int, np.integer)):
        return int(np.int64(x) % q)
    return (x % q).astype(np.int64)
```

5. Şifreleme

encrypt_int() fonksiyonu bir tam sayı mesajını LWE şemasıyla şifreler. Rastgele bir a vektörü seçilir, Gauss dağılımından e hatası üretilir ve $b = \langle a, s \rangle + \Delta \cdot m + e \bmod q$ hesaplanarak şifreli metin oluşturulur.

```
# ŞİFRELEME
def encrypt_int(m: int, sk: LWESecretKey):
    p   = sk.params
    rng = np.random.default_rng()
    a   = rng.integers(0, p.q, size=p.n, dtype=np.int64)
    e   = int(round(rng.normal(0, p.std_dev * p.q)))
    mu  = p.delta * m
    inner = int(np.sum(a * sk.s) % p.q)
    b   = _mod_q(inner + e + mu, p.q)
    return LWECiphertext(a=a, b=b, params=p)
```

6. Şifre Çözme

decrypt_int() fonksiyonu şifreli metinden faz değerini hesaplar: φ = b − ⟨a, s⟩ mod q. Ardından φ, Δ'ya bölünerek en yakın tam sayıya yuvarlanır ve orijinal mesaj elde edilir.

```
# ŞİFRE ÇÖZME
def decrypt_int(ct: LWECiphertext, sk: LWESecretKey):
    p     = ct.params
    inner = int(np.sum(ct.a * sk.s) % p.q)
    phase = _mod_q(ct.b - inner, p.q)
    return int(round(phase / p.delta)) % (2**p.plaintext_bits)
```

7. Homomorfik İşlemler

homo_add() fonksiyonu iki şifreli metni şifreyi çözmeden toplar. homo_scalar_mul() ise bir şifreli metni skaler bir sabit ile çarpar. Her iki işlem de şifreli alan üzerinde çalışır ve gizli anahtara erişim gerektirmez.

```
# HOMOMORFİK İŞLEMLER
def homo_add(ct1, ct2):
    p = ct1.params
    return LWECiphertext(
        a=_mod_q(ct1.a + ct2.a, p.q),
        b=_mod_q(ct1.b + ct2.b, p.q),
        params=p
    )

def homo_scalar_mul(ct, k):
    p = ct.params
    return LWECiphertext(
        a=_mod_q(ct.a * k, p.q),
        b=_mod_q(ct.b * k, p.q),
        params=p
    )
```

8. Programmable Bootstrapping (Sadeleştirilmiş)

Programmable_bootstrap() fonksiyonu gürültü birikimini gidermek için şifreli metni yeniler. Bu sadeleştirilmiş uygulamada işlem gizli anahtar üzerinden gerçekleştirilmektedir. Gerçek TFHE uygulamalarında bu adım tamamen şifreli alan üzerinde yürütülür.

```
# BOOTSTRAPPING (sadeleştirilmiş)
```

```
def programmable_bootstrap(ct, sk, lut):
    m     = decrypt_int(ct, sk)    # Gürültülü şifre çöz
    m_out = lut[m]                 # LUT dönüşümünü uygula
    return encrypt_int(m_out, sk) # Düşük gürültüyle yeniden şifrele
```

9. Nicemleme ve Geri Ölçekleme

quantize() fonksiyonu gerçek sayı verilerini belirli bir bit derinliğinde tam sayı gösterimine dönüştürür. dequantize() ise tahmin sonucunu orijinal ölçeğe geri çevirir.

```
# NİCEMLEME (QUANTIZATION)
def quantize(data, bits):
    min_val = data.min()
    max_val = data.max()
    scale   = (2**bits - 1) / (max_val - min_val + 1e-9)
    q_data  = np.clip(
                np.round((data - min_val) * scale),
                0, 2**bits - 1).astype(int)
    return q_data, min_val, max_val

def dequantize(q_data, min_val, max_val, bits):
    scale = (max_val - min_val) / (2**bits - 1 + 1e-9)
    return q_data * scale + min_val
```

10. Veri Seti Şifreleme ve Şifre Çözme

encrypt_dataset() fonksiyonu bir veri matrisinin tüm öğelerini LWE şemasıyla şifreler. decrypt_dataset() ise şifreli çıkarım sonuçlarını çözerek geri ölçekleme uygular ve tahmin değerlerini orijinal sayısal aralığa taşır.

```
# VERİ SETİ ŞİFRELEME
def encrypt_dataset(data, sk):
    q_data, min_val, max_val = quantize(data, sk.params.plaintext_bits)
    encrypted = []
    for row in q_data:
        encrypted_row = [encrypt_int(int(x), sk) for x in row]
        encrypted.append(encrypted_row)
    return encrypted, (min_val, max_val)

def decrypt_dataset(enc_data, sk, params):
    min_val, max_val = params
    result = []
    for row in enc_data:
        decrypted_row = [decrypt_int(ct, sk) for ct in row]
        result.append(decrypted_row)
    return dequantize(
               np.array(result), min_val, max_val,
```

```
                sk.params.plaintext_bits)
```