TÜRKİYE CUMHURİYETİ'NİN YÜZÜNCÜ YILI

PRATİK YAPAY ZEKA UYGULAMALARI - II

Editör
Eyyüp Gülbandılar

BİDGE

BİDGE Yayınları

Pratik Yapay Zeka Uygulamaları 2

Editör: Prof. Dr. Eyyüp GÜLBANDILAR

ISBN: 978-625-372-396-5

1. Baskı
Sayfa Düzeni: Gözde YÜCEL
Yayınlama Tarihi: 25.12.2024
BİDGE Yayınları

Bu eserin bütün hakları saklıdır. Kaynak gösterilerek tanıtım için yapılacak kısa alıntılar dışında yayıncının ve editörün yazılı izni olmaksızın hiçbir yolla çoğaltılamaz.

Sertifika No: 71374
Yayın hakları © BİDGE Yayınları
www.bidgeyayinlari.com.tr - bidgeyayinlari@gmail.com
Krc Bilişim Ticaret ve Organizasyon Ltd. Şti.
Güzeltepe Mahallesi Abidin Daver Sokak Sefer Apartmanı No: 7/9 Çankaya / Ankara

İçindekiler

Derin Öğrenme ve Makine Öğrenmesi Algoritmaları Kullanarak Borsa Fiyat Tahmini Yapma 5

Sümeyye NAÇAR 5

Hamza Can KORKMAZ 5

Ayşe Berna ALTINEL 5

Orange DM Aracını Kullanarak Topluluk Öğrenme Tekniklerinin Uygulanması 30

Bashar ALHAJAHMAD 30

Musa ATAŞ 30

Meta-Sezgisel Optimizasyon Yöntemler Üzerine Bir İnceleme 51

Ebubekir SEYYARER 51

Faruk AYATA 51

Kara Kutudan Açıklanabilir Yapay Zekaya 69

Musa ATAŞ 69

Bashar ALHAJAHMAD 69

Parçacık Sürü Optimizayon (PSO) Algoritması 92

UğurTALAŞ 92

Patates Yaprak Görüntüleri İle Hastalık Teşhisi: Vision Transformer Ve VGG16 Derin Öğrenme Yöntemlerinin Karşılaştırması 128

Alper Talha KARADENİZ 128

Brats20 veri seti ile beyin tümörü segmentasyonunda Segnet ve MambaUnet'in derin öğrenme performanslarının karşılaştırılması 153

Rabia UYAR 153

Etkili İlaç Yönetimi için Yenilikçi Bir Mobil Uygulama 170

Samet ÖZER 170

Büşra TAKGİL 170
Makine Öğrenmesi: Yöntemler, Algoritmalar Ve Gerçek Dünya Uygulamaları 187
Şükran YAMAN ATCI 187
Fatma KOSOVALI ÇAVUŞ 187
Zeynep İNEL ÖZKİPER 187
Sara NAGHIBZADEH 187
Daha Yüksek Verimlilik ve Güvenlik için Açık Bankacılık Platformu Geliştirme 210
Zeynep YAVUZ 210
Zeynep Nur SANDIKÇI 210
Önder DEMİR 210
Kazım YILDIZ 210

BÖLÜM I

Derin Öğrenme ve Makine Öğrenmesi Algoritmaları Kullanarak Borsa Fiyat Tahmini Yapma

Sümeyye NAÇAR[1]
Hamza Can KORKMAZ[2]
Ayşe Berna ALTINEL[3]

1. Giriş

Borsa fiyat tahminleri, yatırımcılar ve analistler için önemlidir ve hisse senedi fiyatları birçok parametreye bağlı olarak değişir. Bu parametrelerin analizi ve geçmiş verilerle oluşturulan modeller, gelecekteki fiyat hareketlerini öngörmeyi sağlar (Şişmanoğlu & ark., 2020; Wei, Xu & He, 2020). Ancak, piyasa dinamiklerinin

[1] Marmara Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği, İstanbul/Türkiye, Orcid:0009-0006-3482-0834, sumeyyenacar@marun.edu.tr
[2] Marmara Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği, İstanbul/Türkiye, Orcid: 0009-0006-3700-6571, hamza.can@marun.edu.tr
[3] Marmara Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul/Türkiye, Orcid: 0000-0001-5544-0925, berna.altinel@marmara.edu.tr

belirsizliği ve veri karmaşıklığı nedeniyle bu tahminler zorludur (Shah & ark., 2021).

Bu çalışmada, borsa fiyat tahmini için Evrişimsel Sinir Ağları (CNN), Tekrarlayan Sinir Ağları (RNN), Kapı Özyinelemeli Birimler (GRU), Uzun Kısa Süreli Bellek (LSTM), Çift Yönlü LSTM (BiLSTM) ve Aşırı Gradyan Artırma (XGBoost) gibi yapay zeka ve makine öğrenimi modelleri incelenmiştir. CNN ve RNN gibi derin öğrenme teknikleri, GRU ve LSTM gibi gelişmiş RNN türevleri ile hisse senedi fiyatları tahmin edilmeye çalışılmıştır. XGBoost ise karar ağaçları ve boosting yöntemlerini birleştirerek yüksek doğruluk oranları sağlar (Gumelar, 2020).

2010-2024 yılları arasındaki Amazon hisse fiyatları üzerinde eğitilen modellerin performansları karşılaştırılmıştır. Ayrıca, Yığınlanmış Topluluk Modeli (Stacked Ensemble) kullanılarak modellerin birleştirilmesi ve tahmin performansının iyileştirilmesi hedeflenmiştir. Bu yaklaşım, farklı modellerin güçlü yönlerini birleştirerek daha güvenilir ve doğru tahminler elde etmeyi amaçlar.

Çalışmanın ilk aşamasında, her bir modelin temel mimarisi ve teorik arka planı incelenmiştir. RNN modeli ve bunların gelişmiş türevleri olan GRU, LSTM ve BiLSTM modelleri zaman serileri ve sıralı veriler üzerinde başarılı sonuçlar vermektedir. XGBoost ise, karar ağaçları ve boosting yöntemlerinin birleşimi ile yüksek doğruluk oranlarına ulaşan bir algoritmadır (Gumelar, 2020). Eğitim süreçleri boyunca kullanılan veriler, 2010-2024 yılları arasındaki Amazon hisse fiyatlarından oluşmaktadır. Modellerin performansları, iniş ve çıkış hareketlerini yakalama ve gerçek fiyatlara yakın tahminlerde bulunma açısından karşılaştırılmıştır.

Ayrıca, Yığınlanmış Topluluk Modeli (Stacked Ensemble) öğrenme tekniği kullanılarak modellerin birleştirilmesi ve tahmin performansının iyileştirilmesi sağlanmıştır. Stacked, farklı modellerin çıktılarının bir araya getirilerek bir meta-model oluşturulması ve bu sayede genel tahmin doğruluğunun arttırılması prensibine dayanır. Bu yaklaşımla, bireysel modellerin güçlü yönlerini birleştirerek daha güvenilir ve doğru tahminler elde etmeyi amaçlanmıştır.

Çalışmanın kalan kısmı şu şekilde yapılandırılmıştır. İkinci bölümde bu konu ile ilgili yapılan çalışmalar incelenmiştir. Üçüncü bölümde makalede kullanılan materyal ve yöntemler açıklanmıştır. Dördüncü bölümde gerçekleştirilen deneylerin sonuçları sunulmuş ve yorumlanmıştır. Son bölümde ise elde edilen bulgular özetlenmiş ve makalenin literatüre olan katkıları tartışılmıştır.

2. İlgili Çalışmalar

Hiransha ve arkadaşları (2018), derin öğrenme modellerinin borsa tahminlerinde kullanımını incelemiştir. Çalışmada, Çok Katmanlı Algılayıcı (MLP), RNN, LSTM ve CNN olmak üzere dört farklı derin öğrenme mimarisi kullanılarak Hindistan Ulusal Borsası (NSE) ve New York Borsası (NYSE) verileriyle hisse senedi fiyatları tahmin edilmiştir. Sonuçlar, CNN'in diğer modellerden daha iyi performans gösterdiğini ve her iki borsada da ortak iç dinamikleri anlayabildiğini göstermiştir. Bu bulgular, derin öğrenme modellerinin borsa tahmininde geleneksel ARIMA modelinden daha üstün olduğunu ortaya koymaktadır.

Wei, Xu & He (2020), hisse senedi kapanış fiyatlarının tahmininde derin öğrenme tekniklerinin kullanımını araştırmışlardır.

Özellikle, hisse senedi kapanış fiyatlarını tahmin etmek için LSTM tabanlı bir model geliştirmişlerdir. Çalışmalarında, modelin doğruluğunu artırmak için dikkat (attention) mekanizması eklenmiştir. Deneyler, modelin CSI 300 ve Hang Seng Endeksi veri setlerinde düşük hata oranıyla başarılı sonuçlar verdiğini göstermiştir. Bununla birlikte, modelin daha doğru sonuçlar elde edebilmesi için büyük miktarda veriye ve bu verileri işlemek için yüksek hesaplama gücüne ihtiyaç duyduğu belirtilmiştir.

Ayrıca başka bir çalışmada Akbulut & Adem, (2023) gelişmekte olan ülkelerde finansal araçlar arasındaki etkileşimlerin Borsa İstanbul 100 (BIST 100) endeksinin hareketini önemli ölçüde etkilediğini belirtilmiştir. Yapılan bir çalışmada, BIST 100 endeksinin gelişmekte olan ülkelerdeki finansal araçlarla etkileşiminin tahmin edilmesi için derin öğrenme ve makine öğrenmesi yöntemleri kullanılmıştır. Benzer şekilde, Mukherjee ve arkadaşları (2021) tarafından yapılan bir çalışmada, derin öğrenme algoritmaları olan yapay sinir ağı ve CNN kullanılarak borsa tahmini yapılmıştır.

Althelaya, El-Alfy & Mohammed (2020), finansal zaman serilerinin kısa ve uzun vadeli tahmininde LSTM derin öğrenme mimarilerini değerlendirmiştir. Çalışmada, hem çift yönlü (bidirectional) hem de yığılmış (stacked) LSTM modelleri kullanılarak, bu modellerin performansları basit LSTM ve sığ sinir ağlarıyla karşılaştırılmıştır. S&P500 borsa verileri kullanılarak yapılan deneyler, çift yönlü ve yığılmış LSTM modellerinin, kısa vadeli tahminlerde uzun vadeli tahminlere göre daha iyi performans gösterdiğini ortaya koymuştur. Çalışmanın sonuçları, LSTM tabanlı

derin öğrenme modellerinin finansal zaman serisi tahmininde önemli avantajlar sunduğunu doğrulamaktadır.

Gao, Wang & Zhou (2021), hisse senedi tahmini için optimize edilmiş LSTM ve GRU modellerini karşılaştıran bir çalışma yapmıştır. Çalışmada, yatırımcı duyarlılığı göstergeleri ve finansal veriler gibi çeşitli teknik göstergeler entegre edilerek, LASSO ve PCA yöntemleri ile boyut indirgeme işlemleri gerçekleştirilmiştir. Deneysel sonuçlar, LSTM ve GRU modellerinin her ikisinin de hisse senedi fiyatlarını etkili bir şekilde tahmin edebildiğini göstermiştir. Bu çalışma, derin öğrenme yöntemlerinin finansal zaman serisi tahminlerindeki etkinliğini vurgulamakta ve LSTM ile GRU modellerinin farklı boyut indirgeme yöntemleri altında performanslarını karşılaştırmaktadır.

Selvin ve arkadaşları (2017), CNN, LSTM ve RNN modelleri kullanmış ve bu modellerin hisse senedi fiyat tahmininde etkili olduğu göstermiştir. Özellikle CNN, trend değişimlerini belirlemede en başarılı model olarak bulunmuştur. Bu çalışmada kullanılan kaydırma pencere yaklaşımı, yüksek frekanslı ticaret gibi algoritmik ticaret uygulamaları için kullanılabilir. Sonuç olarak, CNN mimarisinin, hisse senedi piyasalarındaki ani değişiklikleri belirlemede diğer modellerden daha iyi performans gösterdiği bulunmuştur.

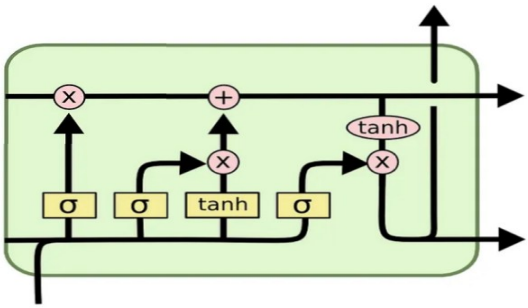
3. Yöntem

3.1 Kullanılan Makine ve Derin Öğrenme Modelleri

3.1.1 LSTM (Uzun Kısa Süreli Bellek)

Derin öğrenme, özellikle zaman serisi analizi ve sıralı veri işleme konularında büyük başarılar elde etmiştir. Bu tür görevlerde

yaygın olarak kullanılan modellerden biri, LSTM ağlarıdır. LSTM'ler, RNN yapısının bir varyasyonu olup, uzun vadeli bağımlılıkları öğrenme kapasitesiyle öne çıkar. Şekil 1'de LSTM şeması görülmektedir.

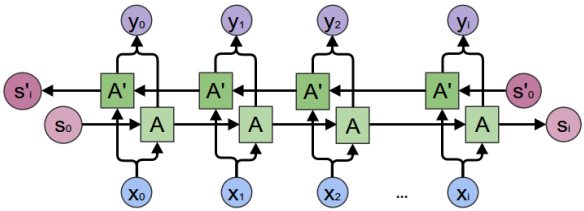


Şekil 1: LSTM Şeması

Kaynak: ResearchGate. (2023). Long short-term memory. (30/06/2024 tarihinde https://www.researchgate.net/figure/Long-short-term-memory_fig2_376444721 adresinden ulaşılmıştır)

3.1.2 BiLSTM (Çift Yönlü Uzun Kısa Süreli Bellek)

BiLSTM modeli, geleneksel LSTM modeline kıyasla iki yönlü bilgi akışı sağlayarak daha zengin ve kapsamlı veri temsilini mümkün kılar. BiLSTM ağları, ileri yönlü ve geri yönlü olmak üzere iki ayrı LSTM katmanından oluşur. Bu çift yönlü yapı, modelin hem geçmiş hem de gelecek bağlamını aynı anda öğrenmesine olanak tanır. İleri yönlü LSTM, veriyi zaman sırasına göre işler. Her bir zaman adımında, önceki zaman adımından gelen bilgi ile mevcut girdiyi kullanarak bir çıktı üretir ve bu çıktıyı bir sonraki zaman adımına iletir. Şekil 2'de BiLSTM şeması görülmektedir.



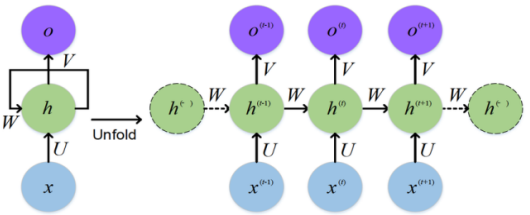
Şekil 2: BiLSTM şeması.

Kaynak: ResearchGate. (2023). Bidirectional recurrent neural network. (30/06/2024 tarihinde https://www.researchgate.net/figure/Bidirectional-Recurrent-Neural-Network-48_fig3_370763497 adresinden ulaşılmıştır.)

Geri yönlü LSTM ise veriyi ters zaman sırasına göre işler. Her bir zaman adımında, bir sonraki zaman adımından gelen bilgi ile mevcut girdiyi kullanarak bir çıktı üretir ve bu çıktıyı bir önceki zaman adımına iletir.

3.1.3 RNN (Tekrarlayan Sinir Ağları)

RNN'ler, özellikle sıralı veri ve zaman serisi analizi gibi görevlerde kullanılan etkili derin öğrenme modelleridir. RNN'ler, verinin zaman içindeki bağımlılıklarını ve dinamiklerini öğrenebilme yetenekleri ile öne çıkar. RNN'ler, her adımda girdiyi alıp işleyerek, önceki adımın çıktısını ve mevcut girdiyi kullanarak bir sonraki adımın çıktısını üreten bir yapıya sahiptir. Bu, modelin zaman içindeki bağımlılıkları öğrenmesini sağlar. RNN'lerin temel bileşeni, gizli durum (hidden state) olarak adlandırılan ve bilgiyi bir zaman adımından diğerine taşıyan nöron katmanıdır. Şekil 3'te RNN şeması görülmektedir.

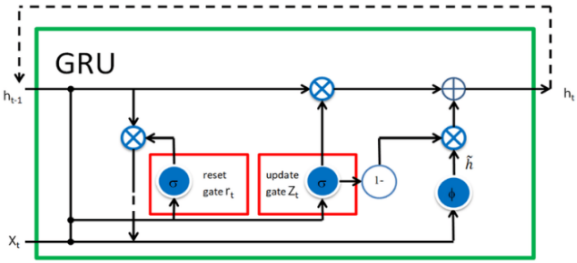


Şekil 3: RNN Şeması.

Kaynak: ResearchGate. (2023). The structure and temporal unfolding of a simple RNN. (30/06/2024 tarihinde https://www.researchgate.net/figure/The-structure-and-temporalunfolding-of-a-simple-RNN-Figure-from-25_fig2_341311377 adresinden ulaşılmıştır.)

3.1.4 GRU (Kapılı Tekrarlayan Birim)

GRU, RNN ailesinin bir üyesi olup, LSTM gibi uzun vadeli bağımlılıkları öğrenme kapasitesine sahip olan bir modeldir. GRU, LSTM'ye benzer bir yapıya sahip olmasına rağmen, daha basit bir mimariyle hesaplama maliyetlerini düşürür ve performansı artırır (Yang, Yu & Zhou, 2020). GRU, LSTM gibi kapı mekanizmaları kullanarak bilgiyi kontrol eder, ancak daha az sayıda kapı içerir ve bu nedenle daha basit bir yapıya sahiptir. GRU'da, iki ana kapı bulunur: güncelleme kapısı (update gate) ve resetleme kapısı (reset gate). Bu kapılar, hücre durumunun güncellenmesini ve korunmasını kontrol eder. Şekil 4'te GRU kaplılarının şeması görülmektedir.



Şekil 4: GRU kapılarının şeması.

Kaynak: ResearchGate. (2023). The diagram of a GRU cell. (30/06/2024 tarihinde https://www.researchgate.net/figure/The-diagram-of-a-GRU-cell_fig3_323570767 adresinden ulaşılmıştır.)

3.1.5 XGBoost (Aşırı Gradyan Artırma)

XGBoost, sınıflandırma ve regresyon problemlerinde yüksek performans gösteren bir makine öğrenimi algoritmasıdır. Güçlü tahmin yetenekleri ve hızıyla öne çıkan XGBoost, özellikle büyük veri setleri üzerinde etkili çalışır. XGBoost, Gradient Boosting Framework'ünün bir uzantısıdır ve modelleme sürecinde bir dizi karar ağacını iteratif olarak eğitir. Her iterasyonda, model hataları minimize etmeyi hedefler ve bu hataların düzeltilmesi için yeni ağaçlar ekler.

3.1.6 Stacked Ensemble (Yığınlanmış Topluluk)

Stacked Ensemble, birbiriyle ilişkili birden fazla tahmin modelini bir araya getirerek daha güçlü bir model oluşturmayı amaçlayan bir ensemble öğrenme yöntemidir. Bu yöntem, farklı özelliklere ve veri dönüşümlerine dayanan çeşitli taban modellerin tahminlerini bir meta-model kullanarak birleştirir. Stacked Ensemble yöntemi, genellikle iki aşamalı bir süreç olarak gerçekleştirilir. İlk aşamada, çeşitli taban modeller (örneğin, XGBoost, LSTM, GRU, RNN) kullanılarak veri seti üzerinde

tahminler yapılır. Bu taban modeller, farklı özelliklere ve hiper parametre ayarlarına sahip olabilir. İkinci aşamada, taban modellerin tahminlerinden oluşan bir özellik matrisi, bir meta-model olarak adlandırılan üst düzey bir model tarafından kullanılır.

Meta-model, taban modellerin tahminlerini girdi olarak alır ve bu tahminlerin bir kombinasyonunu kullanarak nihai tahmini üretir. Bu kombinasyon genellikle bir regresyon modeli veya başka bir öğrenme algoritması ile yapılır.

Meta-öğrenme, birden fazla temel öğrenicinin çıktısından bir meta-öğreniciyi eğitmeyi içerir. Stacking özellikle aynı veri kümesinden farklı bir eğitim algoritması kullanarak birkaç temel öğreniciyi eğitir. Her temel öğrenici, görünmeyen bir veri kümesinde tahminler yapar. Bu ilk model tahminleri daha sonra bir araya getirilir ve bir meta-model olan nihai bir modeli eğitmek için kullanılır (Mienye & Sun, 2022).

3.2 Verilerin İşlenerek Streamlit Kullanılarak Görselleştirilmesi

Streamlit, veri bilimi ve makine öğrenimi uygulamaları geliştirmeyi hızlandıran açık kaynaklı bir Python kütüphanesidir. Veri bilimciler ve mühendisler, minimal kodla kullanıcı dostu arayüzler oluşturabilirler. Hızlı prototipleme sağlar ve veri görselleştirme araçlarıyla entegre çalışır.Tahminler için değişkenler, derin öğrenme ve makine öğrenimi modellerinin sonuçlarını saklar. LSTM, BiLSTM, CNN, RNN, GRU ve XGBoost modelleri ayrı ayrı değerlendirilir. LSTM modeli dört katmanlıdır, her katmanda 50 birim ve Dropout bulunur. Adam optimizasyon algoritması ile eğitim yapılır ve sonuçlar grafik ve tablo olarak sunulur.

BiLSTM modeli de dört katmanlı olup her katmanda 50 birim ve Dropout vardır. CNN modeli ise evrişim ve havuzlama katmanları içerir, ReLU aktivasyonu kullanır. GRU modeli, her katmanda 50 birim ve %20 Dropout ile yapılandırılır, SGD algoritması ile eğitilir.

4. Deney Ortamı, Sonuçlar ve Değerlendirme

4.1 Veri Seti

Borsa analizi, yatırımcılar ve finans uzmanları için doğru kararlar almakta kritik bir rol oynar ve hisse senetlerinin geçmiş performanslarını inceleyerek gelecekteki eğilimleri öngörmeyi hedefler. Modern veri analitiği ve makine öğrenimi teknikleri, bu analizlerin doğruluğunu artırır. Python programlama dili ve zengin kütüphaneleri, borsa analizi projelerinde yaygın olarak kullanılmaktadır. Bu çalışmada, Python'ın yfinance kütüphanesi kullanılarak borsa verileri elde edilecek ve analiz edilecektir.

yfinance, Yahoo Finance API'sine kolay erişim sağlayan bir Python kütüphanesidir. Kullanıcıların hisse senetleri, endeksler, döviz kurları ve diğer finansal varlıklar hakkında veri toplamasına olanak tanır. Bu kütüphane, veri elde etme ve işleme süreçlerini basitleştirerek finansal analizlerin hızlı ve etkili bir şekilde yapılmasını sağlar.

Tablo 1: Veri setindeki sütunların tanımı.

SÜTUN ADI	TANIMI
Tarih (Date)	Hisse senedi fiyatlarının ilgili günün tarihini belirtir.
Açılış (Open)	Belirli bir günün açılış fiyatını temsil eder.
Düşük (Low)	Belirli bir gün içinde düştüğü en düşük fiyatı temsil eder.
Yüksek (High)	Belirli bir gün içinde düştüğü en yüksek fiyatı temsil eder.
Kapanış (Close)	Belirli bir günün kapanış fiyatını temsil eder.
Volüm (Volume)	Belirli bir günde gerçekleşen işlem hacmini temsil eder.

Tablo 1'de belirtilen borsa veri seti sütunları, hisse senedi piyasasında yaygın olarak kullanılan ve yatırımcılara önemli bilgiler sağlayan temel özellikleri temsil eder. Bu sütunlar, yatırımcıların geçmiş performansı analiz etmelerine ve gelecekteki fiyat hareketlerini tahmin etmelerine yardımcı olabilir. Bu sütunlar, yatırımcıların piyasadaki hareketleri anlamalarına ve ticaret stratejileri oluşturmalarına yardımcı olmak için önemli birer araçtır. Veri setindeki bu bilgileri analiz etmek, hisse senedi piyasasında başarılı bir şekilde işlem yapmanın temelini oluşturabilir. Bu çalışmada, kapanış sütunu tahmin edilmeye çalışılmıştır.

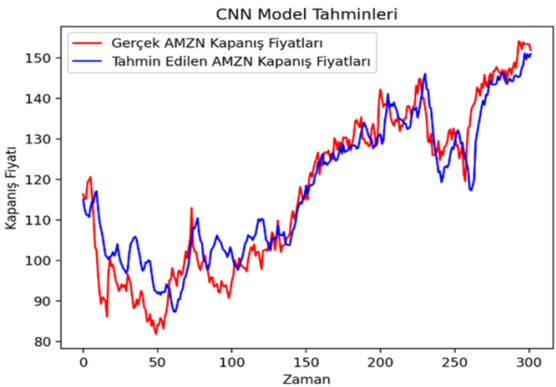
4.2 Sonuçlar ve Değerlendirme

Tüm modeller, 40 epoch çalıştırılarak Amazon şirketine ait kapanış hisse fiyatları tahmin edilmiştir. Bua anlamda Amazon

şirketine ait, 2020.01.02 – 2023.12.29 tarihleri arasındaki veriler %70 eğitim ve %30 test verisi olarak bölünerek model eğitilmiştir.

4.2.1 CNN Modeli ile Gerçekleştirilen Tahminlerin Yorumlanması

Grafik 1'de yer alan grafik gerçek borsa kapanış fiyatını ve CNN modeli ile tahmin edilen hisse kapanış fiyatını göstermektedir. Tahminler 2022.10.18 – 2023.12.29 arası için gerçekleştirilmiştir.



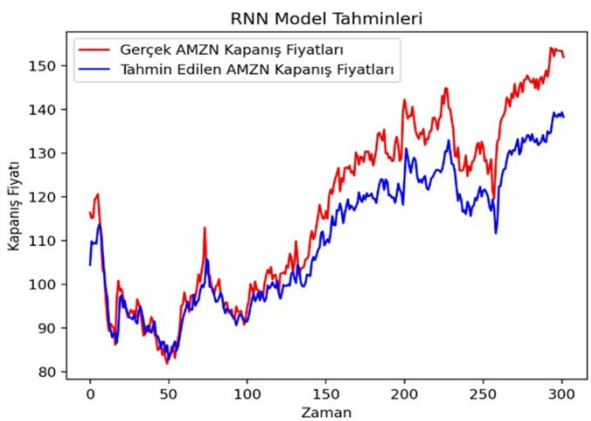
Grafik 1: Amazon hisse fiyatlarının CNN modeli ile tahmin edilmesi.

Grafik 1'de görüldüğü üzere, CNN modeli Ekim 2022 ve Aralık 2023 arasındaki verileri başarılı şekilde tahmin etmiştir. Ancak ilk 140 günü tahin ederken, gerçek fiyatlara göre daha yukarıdan bir tahmin yaptığı görülmüştür. Ancak yukarıdan bir tahmin yapmasına rağmen iniş ye yükselişleri doğru tahmin ettiği görülmektedir. Ayrıca, iniş ve yükselişleri birkaç gün geriden takip ettiği görülmektedir. Ancak yukarıdan bir tahmin yapmasına rağmen iniş ye yükselişleri doğru tahmin ettiği görülmektedir. Bu durumlara rağmen, CNN modelinin Ekim 2022 ve Aralık 2023 tarihlerindeki hisse senedi fiyatlarını tahmin etmede başarılı olduğu söylenebilir.

Ek olarak, CNN modeli LSTM ve Bidirectional LSTM modellerine kıyasla çok daha hızlı çalışmaktadır. Ancak bu modellere kıyasla, daha fazla oranda eğitim verisine ihtiyaç duymaktadır.

4.2.2 RNN Modeli ile Gerçekleştirilen Tahminlerin Yorumlanması

Grafik 2'de yer alan grafik gerçek borsa kapanış fiyatını ve RNN modeli ile tahmin edilen hisse kapanış fiyatını göstermektedir. Tahminler 2022.10.18 – 2023.12.29 arasındaki hisse fiyatları için gerçekleştirilmiştir.



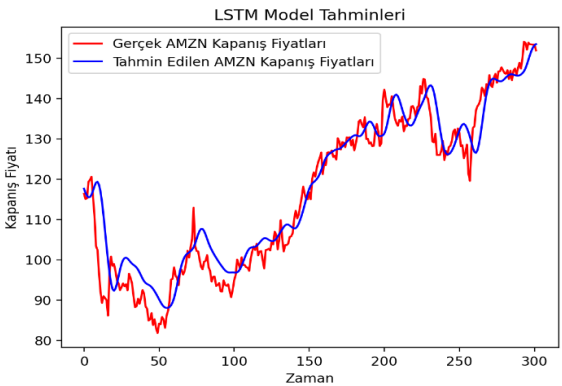
Grafik 2: Amazon hisse fiyatlarının RNN modeli ile tahmin edilmesi.

Grafik 2, 2022.10.18 - 2023.12.29 arasındaki gerçek Amazon kapanış fiyatlarını ve RNN modeliyle tahmin edilen fiyatları göstermektedir. Grafik 2, RNN modelinin genel olarak başarılı tahminlerde bulunduğunu, ancak 150 günden sonra tahminlerin ortalama 5-10 birim düşük kaldığını ortaya koymaktadır. Fiyat artış ve düşüşleri iyi tahmin edilmiştir, ancak RNN'ler uzun vadeli bağımlılıkları öğrenmede zorluk yaşamaktadır. Bu, gradyanların

kaybolması veya patlaması gibi sorunlardan kaynaklanır. Bu problemler, veri miktarı arttıkça tahminlerin kötüleşmesine neden olur. LSTM ve GRU gibi geliştirilmiş RNN türleri, uzun vadeli bağımlılıkları daha iyi öğrenme yetenekleri ile bu sorunları azaltabilir. Buna rağmen, RNN modeli Ekim 2022 - Aralık 2023 döneminde Amazon hisse fiyatlarını genel anlamda iyi tahmin etmiştir, ancak veri miktarı arttıkça doğruluk oranı düşmüştür.

4.2.3 LSTM Modeli ile Gerçekleştirilen Tahminlerin Yorumlanması

Grafik 3'te yer alan grafik gerçek borsa kapanış fiyatını ve LSTM modeli ile tahmin edilen hisse kapanış fiyatını göstermektedir. Tahminler 2022.10.18 – 2023.12.29 arasındaki kapanış hisse fiyatları için gerçekleştirilmiştir.



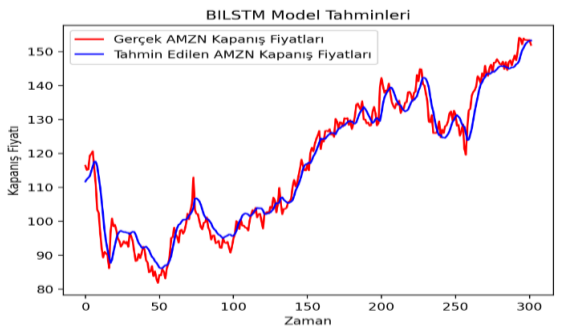
Grafik 3: Amazon hisse fiyatlarının LSTM modeli ile tahmin edilmesi.

Grafik 3'te görüldüğü üzere, LSTM modeli Ekim 2022 ve Aralık 2023 arasındaki verileri başarılı şekilde tahmin etmiştir. Ancak diğer modellere kıyasla, iniş ve yükselişleri tahmin etmede

daha başarısız olduğu görülmektedir. Her ne kadar ani yükseliş ve inişlere diğer modeller kadar iyi tepki vermede daha başarısız olsa da genel anlamda LSTM, fiyat trendini diğer modellere kıyasla daha iyi takip etmiştir. Ayrıca, LSTM'nin büyük verilerde RNN ve CNN modelleri ile kıyaslandığında daha başarılı tahminler yapabildiği gözlemlenmiştir. Bu anlamda daha büyük veri setlerinde daha tutarlı sonuçlar elde edilebilmektedir. Sonuç olarak, LSTM modelleri, özellikle büyük ve uzun dönemli veri setlerinde, fiyat trendlerinin genel doğruluğunu sağlamak için önemli avantajlar sunmaktadır. Buna karşın, LSTM'nin karmaşıklığı ve yüksek hesaplama maliyetleri, özellikle daha düşük veri miktarlarında veya daha kısa zaman dilimlerinde, daha basit modellerin tercih edilmesine yol açabilir.

4.2.4 BiLSTM Modeli ile Gerçekleştirilen Tahminlerin Yorumlanması

Grafik 4'te yer alan grafik gerçek borsa kapanış fiyatını ve BiLSTM modeli ile tahmin edilen hisse kapanış fiyatını göstermektedir. Tahminler 2022.10.18 – 2023.12.29 arasındaki kapanış hisse fiyatları için gerçekleştirilmiştir.



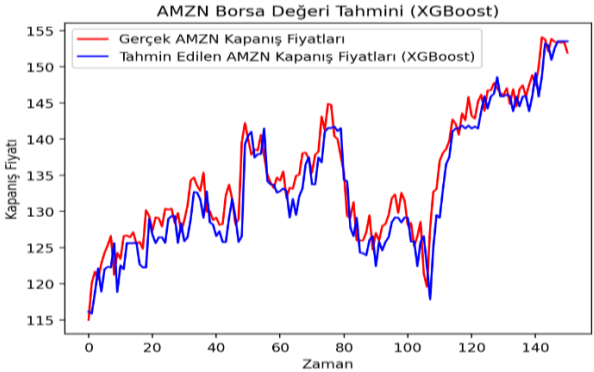
Grafik 4: Amazon hisse fiyatlarının BiLSTM modeli ile tahmin edilmesi.

Grafik 4’te görüldüğü üzere, BiLSTM modeli Ekim 2022 ve Aralık 2023 arasındaki verileri başarılı şekilde tahmin etmiştir. Ayrıca, fiyatlardaki artış ve yükselişleri LSTM modeli ile kıyaslandığında daha iyi tahmin edebildiği görülmüştür. BiLSTM modelleri, standart LSTM modellerinden farklı olarak, veriyi iki yönlü olarak işler. Ancak, BiLSTM’nin karmaşık yapısı, modelin diğer modellere kıyasla daha yavaş çalışmasına sebep olmaktadır. BiLSTM modelleri, çift yönlü bilgi işleme yetenekleri sayesinde daha doğru tahminler üretebilmesi de bu durum modelin hesaplama yükünü artırır.

Sonuç olarak, BiLSTM modeli her ne kadar, düşük çalışma hızına ve yüksek maliyete sahip olsa da veriyi iki yönlü izlemesi sayesinde, Amazon kapanış hisse senedi fiyatlarını başarılı şekilde tahmin etmiştir.

4.2.5 XGBoost Modeli ile Gerçekleştirilen Tahminlerin Yorumlanması

Grafik 5’te yer alan grafik gerçek borsa kapanış fiyatını ve XGBoost modeli ile tahmin edilen hisse kapanış fiyatını göstermektedir. Tahminler 2022.10.18 – 2023.12.29 arasındaki kapanış hisse fiyatları için gerçekleştirilmiştir.

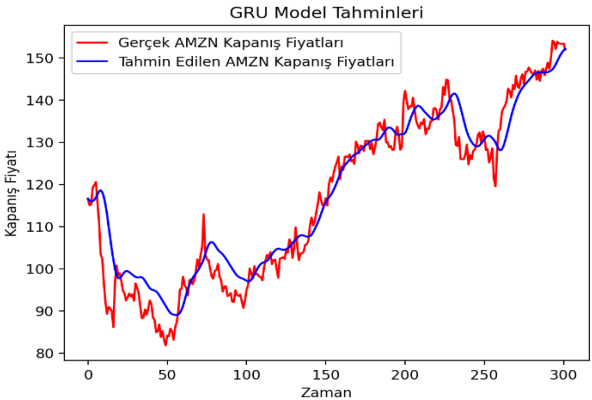


Grafik 5: XGBoost ile alınan tahmin sonuçları.

Grafik 5'te görüldüğü üzere, XGBoost modeli Ekim 2022 ve Aralık 2023 arasındaki verileri başarılı şekilde tahmin etmiştir. Ancak modelin, bazı yükseliş ve artışları tahmin ederken zorlandığı görülmektedir. Ayrıca, XGBoost modelinde, eğitim verisinin oranı azaltıldığında tahminlerin tutarsızlaştığı gözlemlenmiştir. Bununla birlikte, XGBoost modeli büyük veri setlerinde daha etkili tahminler yapmaktadır.

4.2.6 GRU Modeli ile Gerçekleştirilen Tahminlerin Yorumlanması

Grafik 6'da yer alan grafik gerçek borsa kapanış fiyatını ve GRU modeli ile tahmin edilen hisse kapanış fiyatını göstermektedir. Tahminler 2022.10.18 – 2023.12.29 arasındaki hisse fiyatları için gerçekleştirilmiştir.



Grafik 6: Amazon hisse fiyatlarının GRU modeli ile tahmin edilmesi.

Grafik 6'da görüldüğü üzere GRU modeli, Amazon hissesinin kapanış fiyatı tahminlerinde başarılı olmuş ancak tahmin grafiği, diğer modellerin grafiğine kıyasla anlık fiyat değişimlerini daha yüzeysel bir şekilde yansıtmıştır. GRU, RNN ailesine ait bir model olup, LSTM modeline benzer şekilde dizisel veriler üzerinde çalışmak için geliştirilmiştir. Ancak GRU, LSTM'den daha basit ve daha hızlı bir yapıya sahiptir. GRU modelleri, özellikle büyük veri setlerinde ve gerçek zamanlı uygulamalarda tercih edilir. Ancak, daha basit yapısı, modelin bazı karmaşık desenleri yakalama yeteneğini sınırlayabilir. Bu nedenle, GRU modeli anlık fiyat değişimlerini yakalama konusunda diğer daha karmaşık modeller kadar başarılı olmayabilir.

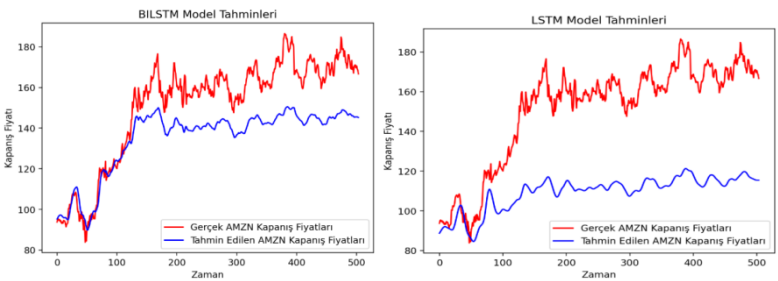
Sonuç olarak, GRU modelinin yapısı ve basit mimarisi, Amazon hissesinin kapanış fiyatı tahminlerinde genel trendleri yakalamada başarılı olurken, anlık fiyat değişimlerini yüzeysel tahmin etmesine neden olmuştur. Bu durum, GRU modelinin

hesaplama maliyetlerini düşürmek için sadeleştirilmiş bir mimariye sahip olmasından kaynaklanmaktadır.

4.2.7 Covid-19 Dönemi için Yapılan Tahminlerin Yorumlanması

Elde edilen sonuçlar geliştirilen modellerin, hisse kapanış fiyatlarını tahmin etmede oldukça etkili olduğunu göstermektedir. Ancak, borsa sosyal olaylara oldukça bağımlıdır ve toplumda aniden ortaya çıkan olağanüstü durumlar sonucu, sıradışı yükseliş ve artışların yaşanması muhtemeldir. Bu anlamda, 2020 yılının başlarında ortaya çıkan Covid-19 pandemisi önemli bir örnek oluşturmaktadır.

Grafik 8'de Covid-19 dönemi için gerçekleştirilen BiLSTM ve LSTM tahminleri görülmektedir. BiLSTM ve LSTM tahminleri, 2018-2022 yılları arasındaki verilerin %50 eğitim ve %50 test olarak bölünmesi ile gerçekleştirilmiştir. Grafik 8'de, BiLSTM ve LSTM tahminleri görülmektedir. Bu sonuçlar incelendiğinde, Her nekadar BiLSTM modeli, Covid-19 pandemi dönemini LSTM modeli ile kıyaslandığında daha başarılı tahmin etmiş olsa da genel anlamda fiyatları tahmin etmede başarısız olmuştur.



Grafik 8: Covid-19 dönemi için gerçekleştirilen BiLSTM ve LSTM tahminleri.

Tüm bu sonuçlar, tahminlerin ancak tutarlı veri setlerinde başarılı olduğunu ancak, derin öğrenme ve makine öğrenmesi ile gerçekleştirilen regresyon tahminlerinin, olağandışı olaylar sonucu gerçekleşen ani değişimleri tahmin etmede başarısız kaldığını göstermektedir.

5. Sonuç

Yapılan analizler sonucunda RNN'lerin uzun vadeli bağımlılıkları öğrenmede zorluk yaşadığı gözlemlenmiştir. Bu sorun, gradyanların kaybolması veya patlaması gibi problemlerden kaynaklanır. RNN ile yapılan hisse fiyat tahminlerinde, veri miktarı arttıkça tahminlerin kötüleştiği görülmüştür.

LSTM ve BiLSTM modellerinin genel anlamda, hisse fiyatlarını tahmin etmede oldukça etkili oldukları görülmüştür. Özellikle BiLSTM'nin veriye hem ileri hem de geriye doğru bakması, yapılan tahminlerin daha doğru olmasını sağlamıştır.

CNN modeli genel anlamda hem iyi tahminlerde bulunmuş hem de LSTM ve BiLSTM algoritmalarına göre çok daha hızlı çalıştığı gözlemlenmiştir. Ancak bu modellere kıyasla, daha fazla oranda eğitim verisine ihtiyaç duymaktadır.

GRU modeli, LSTM'ye benzer bir yapıya sahip olmasına rağmen, daha basit bir mimariye sahip olduğu için fiyatları tahmin ederken ani iniş ve yükselişleri tahmin etmede diğer modellerin gerisinde kalabilir. Ancak, daha basit bir mimariye sahip olduğu için daha hızlı çalışır.

XGBoost modelinde, eğitim verisinin oranı azaltıldığında, tahminlerin tutarsızlaştığı gözlemlenmiştir. XGBoost modeli büyük veri setlerinde daha etkili tahminler yapmaktadır.

Son olarak, modellerin Covid-19 dönemini tahmin etmede yetersiz kaldığı görülmüştür. Bu, regresyon tahminlerinin, toplumsal olaylara karşı savunmasız olmasından kaynaklanmıştır. Genel olarak, geliştirilen modellerin, uzun vadeli tahminlerde oldukça iyi sonuçlar verdiği görülmüştür. Ancak sonuçlar veri setlerinin boyutuna ve dönemsel fiyat artışlarına duyarlı olabilmektedir.

Kaynaklar

Akbulut, S., & Adem, K. (2023). Derin öğrenme ve makine öğrenmesi yöntemleri kullanılarak gelişmekte olan ülkelerin finansal enstrümanlarının etkileşimi ile Bist 100 tahmini. *Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi,* 52 (1), 15-01.

Althelaya, K. A., El-Alfy, E.-S. M., & Mohammed, S. (2020). Evaluation of Bidirectional LSTM for Short- and Long-Term Stock Market Prediction. *In Proceedings of the International Conference on Information and Communication Systems (ICICS).*

Gao, Y., Wang, R., & Zhou, E. (2021). Stock Prediction Based on Optimized LSTM and GRU Models. *Scientific Programming.*

Gumelar, A. B. (2020). Boosting the Accuracy of Stock Market Prediction using XGBoost and Long Short-Term Memory. *In 2020 International Seminar on Application for Technology of Information and Communication (iSemantic)* ,609-613.

Hiransha, M., Gopalakrishnan, E., Menon, V. K., & Soman, K. P. (2018). NSE Stock Market Prediction Using Deep-Learning Models. *In Proceedings of the International Conference on Computational Intelligence and Data Science (ICCIDS).*

Mienye, D., & Sun, Y. (2022). A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects. *IEEE Access.*

Mukherjee, S., Sadhukhan, B., Sarkar, N., Roy, D., & De, S. (2021). Stock market prediction using deep learning algorithms. *CAAI Transactions on Intelligence Technology,* 8.

ResearchGate. (2023). Bidirectional recurrent neural network. (30/06/2024 tarihinde https://www.researchgate.net/figure/Bidirectional-Recurrent-Neural-Network-48_fig3_370763497 adresinden ulaşılmıştır.)

ResearchGate. (2023). Long short-term memory. (30/06/2024 tarihinde https://www.researchgate.net/figure/Long-short-term-memory_fig2_376444721 adresinden ulaşılmıştır.)

ResearchGate. (2023). The diagram of a GRU cell. (30/06/2024 tarihinde https://www.researchgate.net/figure/The-diagram-of-a-GRU-cell_fig3_323570767 adresinden ulaşılmıştır.)

ResearchGate. (2023). The structure and temporal unfolding of a simple RNN. (30/06/2024 tarihinde https://www.researchgate.net/figure/The-structure-and-temporal-unfolding-of-a-simple-RNN-Figure-from-25_fig2_341311377 adresinden ulaşılmıştır.)

Selvin, S., Ravi, V., Gopalakrishnan, E. A., Menon, V., & Soman, K. P. (2017). *Stock price prediction using LSTM, RNN and CNN-sliding window model* içinde (ss. 1643-1647).

Şişmanoğlu, G., Koçer, F., Önde, M. A., & Şahingöz, Ö. K. (2020). Derin Öğrenme Yöntemleri ile Borsada Fiyat Tahmini. *BEÜ Fen Bilimleri Dergisi BEU Journal of Science,* 9.

Wei, J., Xu, Q., & He, C. (2020). Deep learning of predicting closing price through historical adjustment closing price. *Procedia Computer Science.*

Yang, S., Yu, X., & Zhou, Y. (2020). LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review

Dataset as an Example. *In 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*. 98-101.

BÖLÜM II

Orange DM Aracını Kullanarak Topluluk Öğrenme Tekniklerinin Uygulanması

Bashar ALHAJAHMAD [1]
Musa ATAŞ [2]

Giriş

Makine öğrenimi alanında, pek çok model geliştirilmiş ve yeni varyantlar keşfedilmeye devam etmektedir. Bu durum, "Hangi algoritma en iyisidir?" veya "Belirli bir problem bağlamı veya veri seti için hangi algoritma daha uygundur?" gibi önemli soruları gündeme getirmektedir. Ancak, beklenildiği gibi, bu sorulara net ve basit bir cevap yoktur (Mienye & Sun, 2022). Beş algoritmanın altı farklı veri seti üzerindeki performansını karşılaştıran ilginç bir çalışma, tüm veri setlerinde en iyi performansı gösteren bir algoritma

[1] Dr. Öğr. Üyesi, Siirt Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Siirt/Türkiye, Orcid: 0009-0009-3455-7206, bashar.aptech@gmail.com
[2] Prof. Dr., Siirt Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Siirt/Türkiye, Orcid: 0000-0002-9406-0076, hakmesyo@gmail.com

olmadığını ortaya koymuştur (Seni & Elder, 2010). Bu gözlem, iki veya daha fazla modelin tahminlerinin birleştirildiği Ensemble Learning (topluluk öğrenmesi) yönteminin kullanılmasına yol açmıştır. Topluluk modelleri, özellikle bağımsız ve çeşitli modellerin tahminlerinin bir araya getirilmesiyle daha yüksek doğruluk sağlayabilmektedir (Surowiecki, 2005).

Makine öğrenimi alanında en öne çıkan iki yaklaşım, topluluk öğrenme ve derin öğrenme yöntemleridir (Mohammed & Kora, 2021; Goel ve ark., 2022; Zhou ve ark., 2021). Topluluk öğrenme, birden fazla temel öğreniciyi (base learner) eğiterek bu öğrenicilerin tahminlerini birleştirme yoluyla daha yüksek performans ve genelleme yeteneği elde etmeyi amaçlar (Mienye, Sun & Wang, 2020). Bu yaklaşımın temeli, makine öğrenimi modellerinin sınırlamalarını ve hata yapma olasılıklarını fark etmeye dayanır. Bu nedenle topluluk öğrenimi, farklı modellerin güçlü yönlerinden yararlanarak sınıflandırma performansını artırmayı hedefler. Makine öğrenimi algoritmalarının genellikle yüksek varyans (high variance), yüksek yanlılık (high bias) ve düşük doğruluk (low accuracy) gibi sınırlamaları vardır (Mishra ve ark., 2022; Sun ve ark., 2021). Yapılan çalışmalar, topluluk modellerinin genellikle bireysel makine öğrenimi modellerinden daha yüksek doğruluğa ulaştığını ortaya koymuştur (Brown, 2010).

Topluluk öğrenme yöntemleri, genellikle **Boosting** (güçlendirme) ve **Bagging** (torbalama) olarak iki ana kategoriye ayrılır (Nguyen ve ark., 2021).

- **Boosting algoritmaları**: Gradient Boosting, XGBoost ve AdaBoost gibi yöntemleri içerir. Bu algoritmalar, ardışık bir şekilde zayıf öğrenicileri güçlendirmeye odaklanır.

- **Bagging algoritmaları**: Random Forest ve Extra Trees gibi yöntemlerle temsil edilir. Bu teknikler, bağımsız ve paralel modeller üreterek performansı artırmayı hedefler.

Bu algoritmalar, makine öğreniminin birçok farklı uygulamasında yüksek performans göstermiştir (Mohammadi ve ark., 2022; Esenogho ve ark., 2022).

Bu bölüm, topluluk öğrenme yöntemlerine ve algoritmalarına odaklanarak, Bagging ve Boosting arasındaki temel farkları açıklamayı amaçlamaktadır. Özellikle, Orange DM (Orange Veri Madenciliği) aracı kullanılarak yaygın olarak kullanılan topluluk algoritmalarının uygulanmasına (Random Forest, AdaBoost ve Gradient Boosting) odaklanılacaktır. Bu inceleme, topluluk öğrenmeyi ve çeşitli topluluk algoritmalarını anlamak isteyen makine öğrenimi araştırmacıları ve uygulayıcıları için hazırlanmıştır.

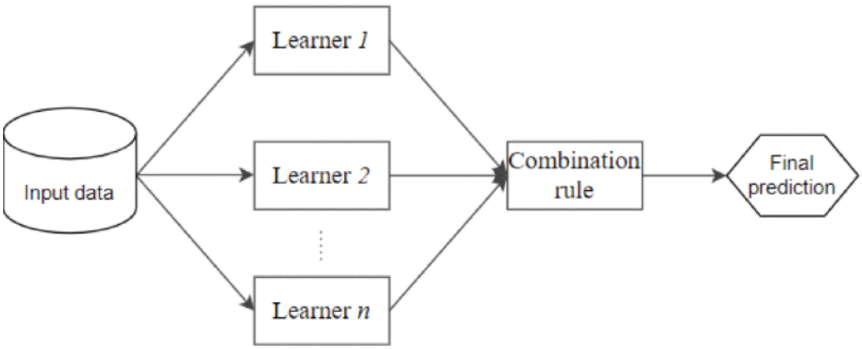
Topluluk Öğrenimine Genel Bakış

Bu bölüm, topluluk öğrenimine genel bir bakış sunarak, topluluk yöntemlerinin yapı taşlarını, baz öğrenicilerin nasıl birleştirileceğini ve topluluk model seçim sürecini açıklamaktadır (Alhajahmad & Ataş, 2023). Çoğu araştırma, topluluk öğreniminin temellerine ilişkin olarak 1979 yılında Dasarathy ve Sheela tarafından yazılan makaleyi önemli bir referans olarak kabul etmektedir (Dasarathy & Sheela, 1979). Bu yazarlar, birden fazla bileşen sınıflandırıcısı kullanarak özellik uzayını bölen bir yöntem sunmuşlardır. 1990 yılında Hansen ve Salamon (Hansen & Salamon, 1990), benzer yapay sinir ağı sınıflandırıcılarından oluşan bir topluluk kullanmanın, tek bir sınıflandırıcıya kıyasla üstün tahmin performansı sağladığını göstermiştir. Aynı yıl içinde Schapire

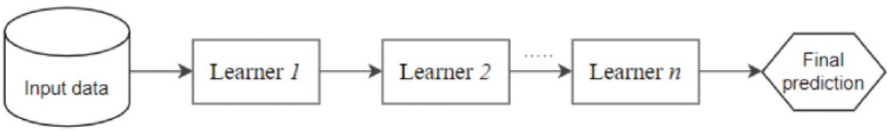
(Schapire, 1990), zayıf bir sınıflandırıcıyı güçlü bir sınıflandırıcıya dönüştürmek için geliştirilmiş bir yöntem olan Boosting tekniğini önermiştir. Bu teknik, günümüzde kullanılan güçlü algoritmalar olan AdaBoost, Gradient Boosting ve XGBoost'un temelini oluşturmuştur (Polikar, 2012).

Paralel topluluklar, baz öğrenicileri bağımsız olarak eğitir ve ardından bu öğrenicilerin tahminlerini birleştirici bir fonksiyonla (Combiner) bir araya getirmektedir. Paralel topluluk yöntemine popüler bir örnek, bagging ve bunun genişletilmiş hali olan random forest algoritmasıdır (Liu, Gegov & Cocea, 2016). Paralel yöntemler, baz modellerin bağımsız olarak oluşturulmasına izin vererek topluluk üyeleri arasında çeşitliliği teşvik etmektedir. Bunun aksine, ardışık topluluklar, baz modelleri bağımsız olarak eğitmemektedir. Bunun yerine, modeller ardışık olarak eğitilir ve her bir iterasyondaki model, önceki modelin yaptığı hataları düzeltmeye çalışır. Boosting algoritması, ardışık topluluklara iyi bir örnektir (Ravichandran & ark., 2021).

Ayrıca, paralel topluluk yöntemleri, baz öğrenicilerin homojenliğine göre homojen veya heterojen olarak sınıflandırılabilir. Homojen topluluklar, aynı makine öğrenimi algoritması kullanılarak inşa edilen modellerden oluşurken, heterojen topluluklar farklı algoritmalarla oluşturulmuş modeller içermektedir (Sabzevari, Martínez-Muñoz & Suárez, 2022)(Bian & ark., 2020). Şekil 1 ve Şekil 2, paralel ve ardışık topluluk öğrenme yaklaşımlarını gösteren blok diyagramları sunmaktadır.



Şekil 1: Paralel topluluk öğrenmesinin blok diyagramı.



Şekil 2: Sıralı topluluk öğrenmesinin blok diyagramı.

Topluluk öğrenme tekniklerinin başarısı, büyük ölçüde temel öğrenicilerin doğruluğuna ve çeşitliliğine bağlıdır (Li, Yu & Zhou, 2012). Bir model, görülmemiş örnekler üzerinde başarılı tahminler yapabiliyorsa "doğru" kabul edilir. Farklı modellerin, aynı veri seti üzerindeki hatalarının farklı olması durumunda, bu modeller "çeşitli" olarak değerlendirilir (Bian ve ark., 2020). Çeşitlilik, bir topluluk içindeki temel öğreniciler arasındaki farkı ifade eder ve bu faktör, topluluk modellerinin başarısında kritik bir rol oynar (Alshdaifat, Al-Hassan & Aloqaily, 2021).

Topluluk Öğrenme Yöntemleri

Ağaç tabanlı (tree-based) topluluk modelleri, birden fazla modelin nasıl üretildiğine bağlı olarak iki temel yaklaşıma ayrılmaktadır: paralel öğrenme ve ardışık öğrenme.

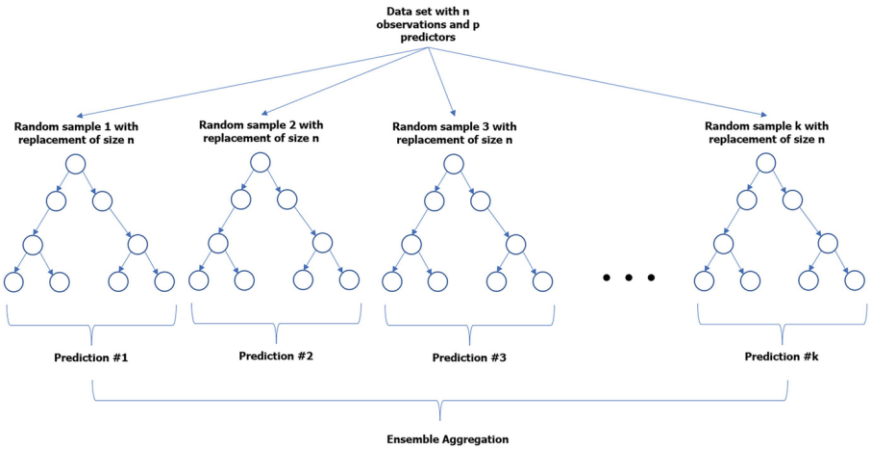
Paralel öğrenme: Bu yaklaşımda, bağımsız ve güçlü (ya da karmaşık) modeller oluşturulur. Paralel yapının temel avantajı, modellerin bağımsız olmaları sayesinde hataların ortalamasını alarak genelleme yeteneğini artırmasıdır. Bagging ve Random Forests, paralel topluluk yöntemlerine örnek olarak gösterilebilir.

Ardışık öğrenme: Bu yöntemde, zayıf modeller ardışık olarak üretilir ve her bir aşamada yapılan hatalar düzeltilerek sonraki modele aktarılır. Bu süreçte, bir modelin hataları, bir sonraki modelin hedef değişkenine dönüşür. AdaBoost, Gradient Tree Boosting ve XGBoost, ardışık topluluk yöntemlerine örnek olarak verilebilir (Acito, 2023).

1) **Bagging** :Bagging (Bootstrap Aggregating), bir paralel topluluk yöntemine erken bir örnektir ve birden fazla veri kümesinden, bootstrap örnekleme yoluyla elde edilen sonuçları birleştirerek tahminler üretmektedir (Breiman, 1996). Bootstrap örneklemede, n gözlem içeren bir veri setinden, k defa tekrarlı olarak n büyüklüğünde örnekler, geri koymalı şekilde alınmaktadır. Şekil 12.1, bu süreci göstermektedir. Tekrarlı örnekleme, bazı gözlemlerin belirli bir örneklemede yer almadığı, bazılarının ise birden fazla kez seçildiği anlamına gelmektedir. Regresyon ağaçlarında, k tahminin ortalaması topluluk tahmini haline gelirken, sınıflandırma ağaçlarında çoğunluk oyu ile sınıf tahmini yapılmaktadır.

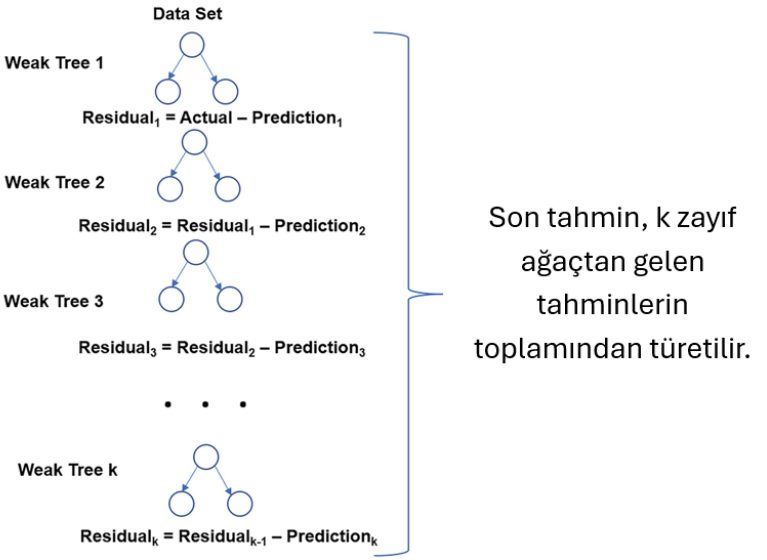
Bir başka paralel topluluk modeli olan Random Forests, üretilen modellerin çeşitliliğini artırmak için ek bir örnekleme aşaması sunmaktadır (Ho, 1998). Her ağacı oluşturmak için tüm kullanılabilir tahmin edici değişkenler (predictors) yerine, rastgele bir değişken örneklemesi yapılmaktadır. Bu nedenle, random forests

iki tür rastgeleleştirme kullanır: (1) gözlemlerin geri koymalı olarak örneklenmesi ve (2) özelliklerin bir alt kümesinin örneklenmesi. Random Forest için kullanılan diyagram, bagging yöntemi için gösterilen Şekil 3'e benzeyecek, ancak her ağacın rastgele seçilen bir özellik alt kümesiyle oluşturulduğu bir fark olacaktır.



Şekil 3: Torbalı (bagged) topluluk ağacı modelinin diyagramı

2) **Boosting**: Boosting, daha büyük, yinelemeli bir karar ağacı oluşturmak için daha küçük karar ağaçlarının ardışık olarak uydurulmasını içerir. Bu daha küçük ağaçlar, genellikle katmanlar (layers) olarak adlandırılır. Her katman, yalnızca birkaç bölünmeye sahip bir karar ağacından oluşur ve bu ağaçlar çok küçük, minimum sayıda bölünmeye sahip olabilir. Boosting süreci, Şekil 4'te gösterilmiştir.



Şekil 4: Güçlendirilmiş ağaç topluluğu modelinin diyagramı

İlk boosting algoritmalarından biri olan AdaBoost, Freund ve Schapire tarafından geliştirilmiştir ve başlangıçta kategorik hedef değişkenler (categorical target variables) için kullanılmıştır (Freund & Schapire,1996). Bununla birlikte, daha sonraki uzantılarla sürekli hedef değişkenler (continuous target variables) de işlenebilir hale gelmiştir. Gradient Boosting Machines (GBM), AdaBoost algoritmasının yaklaşımını geliştirerek, model optimizasyonunda gradyan iniş (gradient descent) yöntemini kullanır (Friedman, 2001). Bu model, zamana bağlı olay projeksiyonları, Poisson regresyonu ve çok terimli sınıflandırma gibi çeşitli bağlamlarda başarıyla uygulanmış ve yaygın olarak kullanılmıştır.

Son yıllarda, XGBoost (eXtreme Gradient Boosting), özellikle Kaggle yarışmalarında kullanımı nedeniyle en popüler topluluk

modelleme tekniklerinden biri haline gelmiştir (Adebayo, 2020). XGBoost, birçok uygulayıcı tarafından "güncel teknoloji" (state-of-the-art) olarak kabul edilmektedir. Esnekliği, özelleştirilebilirliği, açık kaynaklı yapısı, birçok işletim sisteminde çalışabilmesi, aşırı uyumun önlenmesi için düzenleme seçenekleri sunması ve algoritmanın belirli uygulamalara uyarlanması için birçok parametreye sahip olması nedeniyle çok tercih edilmektedir. Ayrıca, XGBoost çoğu karşılaştırılabilir modelden daha hızlı çalışmaktadır. Şekil 5, bagging ve boosting temelli algoritmaların iş akışlarını görsel olarak temsil etmektedir (González & ark., 2020).

Orange Kullanarak Topluluk Öğrenme Algoritmalarının Uygulanması

1. Veri Seti

Bu çalışmada kullanılan veri seti, Kaggle'dan alınan Crop_recommendation veri setidir. Veri seti 2.200 örnekten ve 8 öznitelikten oluşmaktadır (Şekil 6). Veri setinde eksik değer bulunmamaktadır ve sınıflandırma problemleri için oldukça uygundur. Hedef özellik, 22 farklı ürün türünü içermektedir. Çalışmamız için hedef özelliği yalnızca 10 ürün türünü içerecek şekilde daralttık ve bu da veri setinin 1.000 örneğe düşmesine neden oldu.

	Name	Type	Role	Values
1	N	N numeric	feature	
2	P	N numeric	feature	
3	K	N numeric	feature	
4	temperature	N numeric	feature	
5	humidity	N numeric	feature	
6	ph	N numeric	feature	
7	rainfall	N numeric	feature	
8	**label**	C **categorical**	**target**	**apple, banana, coconut, coffee, cotton, grapes, mango, orange, papaya, rice**

Şekil 6: Crop_recommendation veri kümesinin temel özellikleri

2. Orange DM Tool

Bu araştırma, Orange DM yazılımı kullanılarak gerçekleştirilmiştir. Orange, 1997 yılında Janez Demšar ve Blaž Zupan tarafından başlatılan bir veri madenciliği aracının geliştirilmesiyle ortaya çıkmıştır. Geliştirme süreci, başlangıçta Ljubljana Üniversitesi'nde bulunan Yapay Zeka Laboratuvarı'nda başlamış olup, günümüzde her iki kurucu da üniversitenin Biyoinformatik Laboratuvarı'nda çalışmalarını sürdürmektedir. Orange'ın ilk tasarımı, C++ dilinde yazılmış bir kütüphane olarak başlamış ve makine öğrenimi algoritmaları, veri ön işleme, örnekleme ve diğer veri işleme prosedürlerini içermiştir (Alhajahmad, 2023).

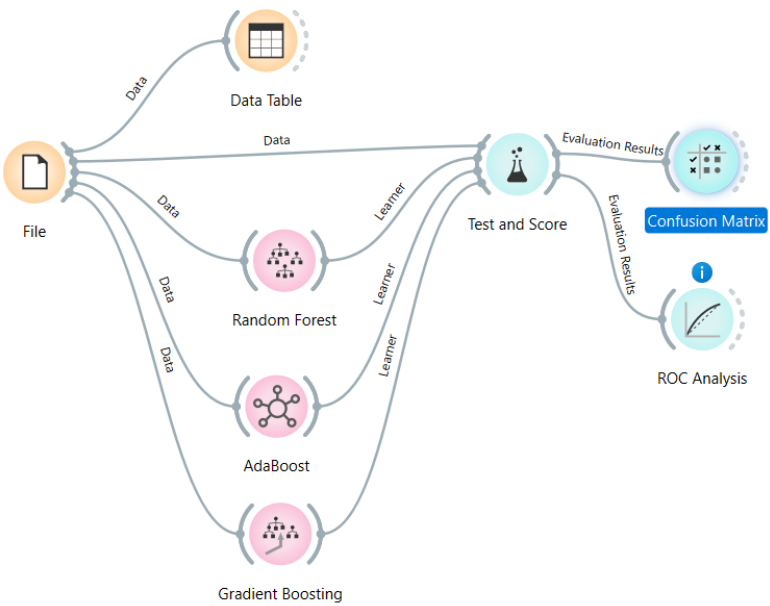
Orange, özellikle farklı ön işleme ve öğrenme algoritmalarının çeşitli kombinasyonlarının çapraz doğrulama yöntemiyle test edilip puanlandığı veri keşfi süreçlerinde kullanılmıştır. Bu bileşenler, ilk aşamada komut satırı arayüzüne sahip programlarda kullanılmıştır. Ancak, bu sınırlı bir yöntem olduğu için Orange geliştiricileri, bu bileşenleri Python ile entegre ederek bir betik arayüzü sağlamaya karar vermişlerdir (Demšar & Zupan, 2012).

Orange, Python dilinde yazılmış, açık kaynaklı ve ücretsiz olarak erişilebilen bir veri madenciliği yazılımıdır (https://orangedatamining.com). Orange, veri keşfi, analiz ve görselleştirme süreçlerinde oldukça etkili bir araç olup, çeşitli deneyler yapmak için güçlü bir platform sunmaktadır. Yenilikçilik, güvenilirlik ve kalite bakımından önemli bir etki yaratan Orange (Pretnar, web sayfası), özellikle Orange Canvas arayüzü sayesinde görsel programlama ortamı sunmakta ve bu sayede kullanıcıların farklı özellikleri düzenli ve yapılandırılmış bir biçimde kullanmasına olanak sağlamaktadır. Şekil 7'de görüldüğü gibi bu özellikler kullanıcıya sunulmaktadır (Ratra & Gulia, 2020).

Şekil 7: Orange'in ana özellikleri

3. Orange Kullanarak Bir Örnek

Çalışmamız kapsamında, Orange ortamında bir iş akışı (workflow) geliştirdik. İlk iş akışı, Şekil 8'de gösterildiği gibi, Crop_recommendation veri seti üzerindeki sınıflandırma problemini çözmeye yönelik olarak tasarlanmıştır. Bu iş akışında özellikle Random Forest, Gradient Boosted ve AdaBoost algoritmalarını uygulamıştık.



Şekil 8: Orange DM'yi kullanan ana iş akışı

Sınıflandırma modellerimizin performansını değerlendirmek için Confusion Matrix bileşenini kullanmıştık. Bu bileşen, karmaşıklık matrisini incelememizi ve önemli istatistiksel metrikler elde etmemizi sağlamıştır. Böylece modellerimizin doğruluğu ve

etkinliği hakkında önemli bilgiler sundu (Şekil 9, Şekil 10 ve Şekil 11).

Predicted

Actual	apple	banana	coconut	coffee	cotton	grapes	mango	orange	papaya	rice	Σ
apple	100	0	0	0	0	0	0	0	0	0	100
banana	0	100	0	0	0	0	0	0	0	0	100
coconut	0	0	100	0	0	0	0	0	0	0	100
coffee	0	0	0	100	0	0	0	0	0	0	100
cotton	0	0	0	0	100	0	0	0	0	0	100
grapes	0	0	0	0	0	100	0	0	0	0	100
mango	0	0	0	0	0	0	100	0	0	0	100
orange	0	0	0	0	0	0	0	100	0	0	100
papaya	0	0	0	0	0	0	0	0	100	0	100
rice	0	0	0	0	0	0	0	0	0	100	100
Σ	100	100	100	100	100	100	100	100	100	100	1000

Şekil 9: Rastgele Orman Modeli İçin Karmaşıklık Matrisi

Predicted

Actual	apple	banana	coconut	coffee	cotton	grapes	mango	orange	papaya	rice	Σ
apple	100	0	0	0	0	0	0	0	0	0	100
banana	0	100	0	0	0	0	0	0	0	0	100
coconut	0	0	100	0	0	0	0	0	0	0	100
coffee	0	0	0	100	0	0	0	0	0	0	100
cotton	0	0	0	0	100	0	0	0	0	0	100
grapes	0	0	0	0	0	100	0	0	0	0	100
mango	0	0	0	0	0	0	100	0	0	0	100
orange	0	0	0	0	0	0	0	100	0	0	100
papaya	0	0	0	0	0	0	0	0	100	0	100
rice	0	0	0	0	0	0	0	0	0	100	100
Σ	100	100	100	100	100	100	100	100	100	100	1000

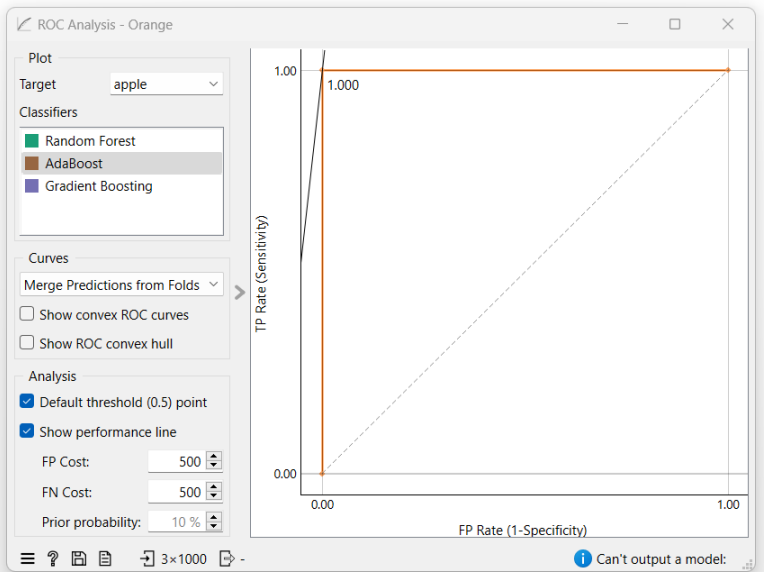
Şekil 10: Adaboost Modeli İçin Karmaşıklık Matrisi

Predicted

Actual	apple	banana	coconut	coffee	cotton	grapes	mango	orange	papaya	rice	Σ
apple	100	0	0	0	0	0	0	0	0	0	100
banana	0	100	0	0	0	0	0	0	0	0	100
coconut	0	0	100	0	0	0	0	0	0	0	100
coffee	0	0	0	100	0	0	0	0	0	0	100
cotton	0	0	0	0	100	0	0	0	0	0	100
grapes	0	0	0	0	0	98	0	0	0	2	100
mango	0	0	0	0	0	0	100	0	0	0	100
orange	0	0	0	0	0	0	0	100	0	0	100
papaya	0	0	0	0	0	0	0	0	100	0	100
rice	0	0	0	0	0	0	0	0	0	100	100
Σ	100	100	100	100	100	98	100	100	100	102	1000

Şekil 11: Şekil 9: Gradient Boosting Modeli İçin Karmaşıklık Matrisi

Ayrıca, model performansını değerlendirmemizi daha da geliştirmek için ROC Curve bileşenini analizimize entegre ettik. Bu bileşen, ROC-AUC eğrileri üretir ve modellerimizin performansını görsel olarak değerlendirmemizi sağlamaktadır (Şekil 12).



Şekil 12: Orange DM kullanılarak ROC Analizi

Şekil 13, Test and Score bileşeni kullanılarak elde edilen farklı performans metriklerinin görsel temsillerini ve Random Forest, AdaBoost ve Gradient Boosting algoritmalarının Crop_recommendation veri seti üzerindeki uygulamalarından elde edilen doğruluk skorlarını sunmaktadır.

Evaluation results for target (None, show average over classes)

Model	AUC	CA	F1	Prec	Recall	MCC
Random Forest	1.000	1.000	1.000	1.000	1.000	1.000
AdaBoost	1.000	1.000	1.000	1.000	1.000	1.000
Gradient Boosting	1.000	0.998	0.998	0.998	0.998	0.998

Şekil 13: Test ve Score Bileşeninden elde edilen doğruluk ve diğer ölçümler

Test and Score bileşeni, hatırlama (recall), kesinlik (precision), özgüllük (specificity), doğruluk (accuracy) gibi temel metriklerin hesaplanmasında ve özellikle AUC-ROC eğrileri oluşturarak model performansına kapsamlı bir bakış sunmada önemli bir rol oynamaktadır.

Şekil 12'deki ROC eğrileri, Random Forest, AdaBoost ve Gradient Boosting algoritmalarının sonuçlarını grafiksel olarak göstermektedir. Bu grafikler, her modelin sınıflandırma görevlerindeki ayrım yapma yeteneklerini ve performansını görsel olarak ortaya koymaktadır.

Şekil 9, Şekil 10 ve Şekil 11'de gösterilen karmaşıklık matrislerinin ayrıntılı incelenmesi, hem Random Forest hem de AdaBoost algoritmalarının sınıflandırma görevinde olağanüstü performans sergilediğini açıkça ortaya koymaktadır. Her iki

algoritma da mükemmel bir şekilde %100 doğruluk elde etmiş ve %0 hata oranı ile sonuçlanmıştır. Buna karşılık, Gradient Boosting algoritması da %98 doğrulukla son derece iyi sonuçlar vermiştir.

Sonuç

Sonuç olarak, bu bölümde, topluluk öğrenme (ensemble learning) tekniklerinin ayrıntıları ele alınmış ve çeşitli uygulamalarda sınıflandırma performansını artırmadaki etkinliklerine vurgu yapılmıştır. Orange DM aracı kullanılarak Random Forest, AdaBoost ve Gradient Boosting gibi yaygın olarak kullanılan algoritmaların sistematik bir şekilde uygulanmasıyla, topluluk yöntemlerinin yüksek doğruluk ve daha iyi genelleme sağlamadaki önemli avantajları gösterilmiştir. Crop Recommendation veri setinden elde edilen etkileyici sonuçlar, birden fazla temel öğreniciyi (base learner) bir araya getirerek bireysel modellerin sınırlamalarını hafifletme potansiyelini vurgulamaktadır. Bu kapsamlı inceleme, makine öğrenimi uygulayıcılarına değerli bilgiler sunmakla kalmayıp, model seçiminde çeşitlilik ve doğruluğun önemini de öne çıkararak, alanda gelecekteki gelişmeler için zemin hazırlamaktadır.

Kaynaklar

Acito, F. (2023). *Predictive analytics with KNIME: Analytics for citizen data scientists*. Switzerland: Springer.

Adebayo, S. (2020). How to solve a problem. Available at. Accessed 29 July 2023.

Alhajahmad, B. (2023*). Orange yazılımı kullanarak veri madenciliği. Mühendislikte İleri ve Çağdaş Çalışmalar 1*. İzmir, Türkiye: Duvar Basımevi, 18-31.

Alhajahmad, B., & Ataş, M. (2023, November 4-5). Boosting Predictive Power: Random Forest and Gradient Boosted Trees in Ensemble Learning. *2nd International Conference on Contemporary Academic Research (ICCAR 2023)*, Konya, Turkey.

Alshdaifat, E., Al-Hassan, M., & Aloqaily, A. (2021). Effective heterogeneous ensemble classification: An alternative approach for selecting base classifiers. *ICT Express*, 7(3), 342-349.

Bian, Y., Wang, Y., Yao, Y., & Chen, H. (2020). Ensemble pruning based on objection maximization with a general distributed framework. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9), 3766-3774.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24, 123-140.

Brown, G. (2010). Ensemble learning. In *Encyclopedia of Machine Learning* (pp. 312-320). Boston, MA: Springer.

Dasarathy, B. V., & Sheela, B. V. (1979). A composite classifier system design: Concepts and methodology. *Proceedings of the IEEE*, 67(5), 708-713.

Demšar, J., & Zupan, B. (2012). Orange: Data mining fruitful and fun. In *Proceedings of the 15th International Multiconference Information Society* – IS, Vol. A.

Esenogho, E., Mienye, I. D., Swart, T. G., Aruleba, K., & Obaido, G. (2022). A neural network ensemble with feature engineering for improved credit card fraud detection. *IEEE Access*, 10, 16400-16407.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning.* Available at. Accessed 29 July 2023.

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 20(5), 1189-1232.

Goel, P., Jain, R., Nayyar, A., Singhal, S., & Srivastava, M. (2022). Sarcasm detection using deep learning and ensemble learning. *Multimedia Tools and Applications*.

González, S., García, S., Del Ser, J., Rokach, L., & Herrera, F. (2020). A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives, and opportunities. *Information Fusion*, 64, 205-237.

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10), 993-1001.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8), 832-847.

Li, N., Yu, Y., & Zhou, Z.-H. (2012). Diversity regularized ensemble pruning. In *Machine Learning and Knowledge Discovery in Databases* (pp. 330-345). Berlin, Germany: Springer.

Liu, H., Gegov, A., & Cocea, M. (2016). Ensemble learning approaches. In *Rule-Based Systems for Big Data: A Machine Learning Approach* (pp. 63-73). Cham, Switzerland: Springer.

Mienye, I. D., & Sun, Y. (2022). A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access*, 10, 99129-99149.

Mienye, I. D., Sun, Y., & Wang, Z. (2020). Improved predictive sparse decomposition method with DenseNet for prediction of lung cancer. *International Journal of Computers*, 19(4), 533-541.

Mishra, S., Shaw, K., Mishra, D., Patil, S., Kotecha, K., Kumar, S., et al. (2022). Improving the accuracy of ensemble machine learning classification models using a novel bit-fusion algorithm for healthcare AI systems. *Frontiers in Public Health*, 10.

Mohammadi, S., Narimani, Z., Ashouri, M., Firouzi, R., & Karimi-Jafari, M. H. (2022). Ensemble learning from ensemble

docking: Revisiting the optimum ensemble size problem. *Scientific Reports*, 12(1), 1-15.

Mohammed, A., & Kora, R. (2021). An effective ensemble deep learning framework for text classification. *Journal of King Saud University - Computer and Information Sciences*.

Nguyen, K. A., Chen, W., Lin, B.-S., & Seeboonruang, U. (2021). Comparison of ensemble machine learning methods for soil erosion pin measurements. *ISPRS International Journal of Geo-Information*, 10(1), 42.

Polikar, R. (2012). Ensemble learning. In *Ensemble Machine Learning: Methods and Applications* (pp. 1-34). Boston, MA: Springer.

Ratra, R., & Gulia, P. (2020). Experimental evaluation of open-source data mining tools (WEKA and Orange). *International Journal of Engineering Trends and Technology*, 68, 30-35.

Ravichandran, T., Gavahi, K., Ponnambalam, K., Burtea, V., & Mousavi, S. J. (2021). Ensemble-based machine learning approach for improved leak detection in water mains. *Journal of Hydroinformatics*, 23(2), 307-323.

Sabzevari, M., Martínez-Muñoz, G., & Suárez, A. (2022). Building heterogeneous ensembles by pooling homogeneous ensembles. *International Journal of Machine Learning and Cybernetics*, 13(2), 551-558.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5(2), 197-227.

Seni, G., & Elder, J. F. (2010). *Ensemble methods in data mining: Improving accuracy through combining predictions*. Morgan & Claypool Publishers.

Sun, Y., Li, Z., Li, X., & Zhang, J. (2021). Classifier selection and ensemble model for multi-class imbalance learning in education grants prediction. *Applied Artificial Intelligence*, 35(4), 290-303.

Surowiecki, J. (2005). *The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations*. Anchor Books.

Zhou, J., Jiang, Z., Chung, F.-L., & Wang, S. (2021). Formulating ensemble learning of SVMs into a single SVM formulation by negative agreement learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(10), 6015-6028.

BÖLÜM III

Meta-Sezgisel Optimizasyon Yöntemler Üzerine Bir İnceleme

Ebubekir SEYYARER[1]
Faruk AYATA[2]

GİRİŞ

Yapay zekanın devrimi, geleneksel veya kesin yöntemler kullanılarak uygulanabilir olmayacak zor ve karmaşık gerçek dünya optimizasyon problemlerine neredeyse optimum çözümler sağlayabilen akıllı problem çözme yöntemlerinin tanıtılmasını sağlar. Kesin yöntemler, bir probleme optimum çözümün bulunmasını sağlarken, yaklaşık yöntemler optimumluğu garanti etmez, ancak daha kısa bir hesaplama süresinde neredeyse optimum bir çözüm elde edebilirler. Yaklaşık yöntemler, sezgisel algoritmalar

[1] Dr. Öğr. Üyesi; Van Yüzüncü Yıl Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü. eseyyarer@yyu.edu.tr ORCID No: 0000-0002-8981-0266
[2] Dr. Öğr. Üyesi; Van Yüzüncü Yıl Üniversitesi Başkale M.Y.O. Bilgisayar Programcılığı Bölümü. farukayata@yyu.edu.tr ORCID No: 0000-0003-2403-3192

ve meta-sezgisel algoritmaları içerir. Sezgisel algoritmalar, probleme bağlı stratejilerdir, yani çözüm sürecinde optimizasyon probleminin özelliklerinden yararlanırlar. Bu nedenle, bir sezgisel yöntem belirli bir probleme göre uyarlanabilir, ancak aynı sezgisel yöntem diğer problemlere uygulanabilir olmayabilir. Öte yandan, meta-sezgisel yöntemler, çok çeşitli problemlere uygulanabilen problemden bağımsız stratejilerdir. Sonuç olarak, optimizasyon problemlerini çözmek için çeşitli sezgisel ve meta-sezgisel teknikler vardır. Meta-sezgisel algoritmalar, 2000 yılından önce geliştirilmişlerse klasik, 2000 yılından sonra geliştirilmişlerse yeni nesil olarak sınıflandırılmaktadırlar. Meta-sezgiler ayrık ve sürekli, kısıtlı ve kısıtsız, tek ve çok amaçlı optimizasyon gibi çok sayıda sayısal problem için kullanılmıştır. (Toaza ve Esztergár-Kiss, 2023:1).

Tipik olarak, gerçek yaşam optimizasyon problemleri zengin ve yumuşak kısıtlamaların yanı sıra düzgün olmayan hedef fonksiyonları içerir, bu da bunların çoğunu NP-Zor optimizasyon problemleri haline getirir. Ek olarak, sıklıkla yüzlerce veya binlerce değişkeni (örneğin müşteriler, ürünler, varlıklar, tesisler vb.) içeren büyük boyutludurlar. Bu senaryoda, kesin optimizasyon yöntemleri sınırlı bir etkinliğe sahiptir ve bu nedenle meta-sezgiseller makul hesaplama sürelerinde neredeyse optimum çözümler üretmek için mükemmel bir alternatif sunar. (Juan vd., 2023:2).

“Meta” terimi günümüzde daha yaygın hale geliyor; genellikle “ötesinde”" veya “daha yüksek seviye” anlamına gelir. Üzerinde mutabık kalınmış bir matematiksel tanım olmasa da, sezgisel algoritmaların sürekli gelişimi genellikle MA olarak adlandırılır. Bir sezgisel algoritma, deneme yanılma yoluyla optimizasyon

sorunlarına kabul edilebilir çözümler üretmek için bir yöntemdir. Zeka yalnızca insanlarda değil, hayvanlarda, mikroorganizmalarda ve karıncalar, arılar ve diğer yaratıklar gibi doğanın diğer küçük unsurlarında da bulunur. Doğa, doğadan esinlenen algoritmalar olarak adlandırılan birçok MA için bir ilham kaynağı görevi görür. Doğa, ışığı uzayda en kısa yoldan hareket ettirmek, herhangi bir canlı organın işlevini en az enerji genişlemesiyle yerine getirmek veya en az yüzey alanına sahip küre olan baloncuklar oluşturmak olsun, tüm görevleri en iyi şekilde gerçekleştirir. Doğal seçilim optimizasyonu destekler. Bu, herhangi bir görevi başarıyla ve zahmetsizce tamamlamanın en verimli yöntemidir. Bu basit kavram, günlük hayatımızda gerçekleştirdiğimiz her türlü işe uygulanabilir. Ancak işletmeler, ulusal güvenlik, geniş alanlarda dağıtım ve bazı yapıların tasarımı gibi büyük ölçekli operasyonlar söz konusu olduğunda, kaynakların düzgün bir şekilde kullanılmasını ve en üst düzeye çıkarılmasını sağlamak için somut bir yönteme veya araca ihtiyacımız olur; bu da operasyon araştırmasına yol açar. Son on yılda, meta-sezgiler operasyon araştırmalarında güçlü bir optimizasyon aracı olarak ortaya çıkmaktadır. Ayrıca, MA'lar esnek, uyarlanabilir ve kapsamlı bir arama kapasitesine sahip oldukları için hesaplamalı zekada daha kritik hale gelmektedir. MA'lar NP-Zor problemlerinde, fikstür ve üretim hücresi tasarımında, yumuşak bilişimde, döviz ticaretinde, robotikte, tıp biliminde, davranış biliminde, fotovoltaik modellerde vb. kullanılır; bu da MA'ların öneminin kanıtıdır. MA'lar doğası gereği rassal(stokastik) olduğundan, optimum çözümün elde edilmesini garanti edemezler. Sonuç olarak, doğal olarak şu soru ortaya çıkar: Değerli bir seçim mi? Bu, kabaca "hiçbir şeyden iyidir" e benzer. Diğerleri başarısız

olduğunda, MA'lar bize tatmin edici bir "bir şey" sağlar. Pratikte, makul bir süre içinde tatmin edici veya uygulanabilir bir çözüme ulaşırız. Algoritmaların çoğu daha düşük boyutlar için test edilmiştir. Bunları daha yüksek boyutlu bir sorun için test etmek ve "boyutluluk laneti" ile başa çıkmak için gerekirse geliştirmek gerekir. Teori ve uygulama arasında önemli bir araştırma boşluğu gösterilmiştir ve bu dikkate alınmalıdır. Keşif ve sömürü çoğu MA'nın temel stratejileri olduğundan, bunları dengelemek başka bir zorluktur (Rajwar vd., 2023:2).

Bu çalışma, güncel literatürlerle meta-sezgisel optimizasyon yöntemlerinin sınıflandırılması üzerine bir inceleme olacaktır. Bu sınıflara ait en popüler yöntemler verilmektedir. Çalışma, yöntemlerin sınıflandırılması ve sonuçlar başlıkları ile devam etmektedir.

YÖNTEMLERİN SINIFLANDIRILMASI

Meta-sezgisel optimizasyon yöntemleri; Sosyal, Fizik, Biyoloji, Kimya, Müzik, Sürü, Spor, Matematik tabanlı olmak üzere sekiz ayrı kategoriye ayrılmaktadırlar (Yang, 2010:18).

1.Sosyal Tabanlı Optimizasyon Algoritmaları

Sosyal tabanlı optimizasyon yöntemleri, insan davranışları ve sosyal etkileşimlerden ilham alarak geliştirilen meta-sezgisel algoritmalardır. Bu algoritmalar, bireylerin veya toplulukların karar alma süreçlerini, iş birliği modellerini ve rekabetçi etkileşimlerini modellemeyi amaçlar. Sosyal optimizasyon algoritmalarında, bireylerin hareketleri, topluluğun genel performansını artırmayı hedefleyen bir şekilde yönlendirilir. Bu yaklaşımlar, özellikle karmaşık optimizasyon problemlerinin çözümünde etkili olmaktadır

(Askeri, 2020). Literatürde yeni önerilen birçok sosyal tabanlı sezgisel optimizasyon algoritması bulunmaktadır. Bunlardan yakın zamanda önerilenler Şekil 1'de listelenmektedir.

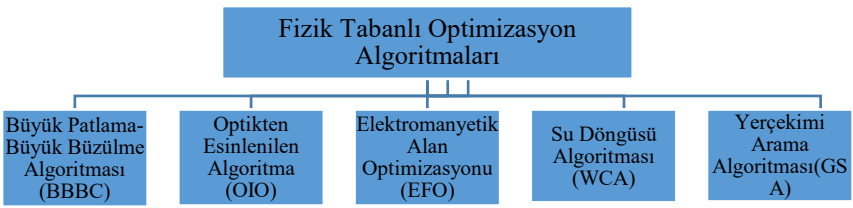
Şekil 1: Sosyal tabanlı optimizasyon algoritmaları (Altunbey ve Alataş, 2015).

Sosyal tabanlı optimizasyon yöntemleri, mühendislikten veri bilimine kadar geniş bir yelpazede kullanılmaktadır. Bu yöntemlerin en büyük avantajlarından biri, karmaşık ve yüksek boyutlu optimizasyon problemlerine uygulanabilir olmalarıdır. Aynı

zamanda, insan davranışlarını ve sosyal etkileşimleri modelledikleri için, bu algoritmalar çoklu hedefli problemler ve kısıtlı optimizasyon görevlerinde etkili çözümler sunar.

2.Fizik Tabanlı Optimizasyon Algoritmaları

Fizik tabanlı optimizasyon algoritmaları, fiziksel süreçler, fenomenler veya fizik yasalarından ilham alarak geliştirilmiş meta-sezgisel yöntemlerdir. Bu algoritmalar, gerçek dünyadaki fiziksel olayları ve bu olayların matematiksel modellerini kullanarak karmaşık optimizasyon problemlerine çözümler sunmayı amaçlar. Fizik tabanlı yöntemler, genellikle keşif ve sömürü aşamalarını dengeli bir şekilde birleştirerek geniş bir çözüm alanında etkili arama yapabilir. Literatür incelendiğinde Fizik tabanlı optimizasyon algoritmalarının bir kısmı Şekil 2'de listelenmektedir.



Şekil 2: Fizik tabanlı optimizasyon algoritmaları (Demirol, 2018)

Fizik tabanlı optimizasyon yöntemleri, özellikle mühendislik, veri bilimi ve robotik gibi alanlarda geniş uygulama alanlarına sahiptir. Bu algoritmalar, gerçek dünyadaki fiziksel süreçlerin dinamiklerini modellemeleri sayesinde birçok farklı problem türü

için uygun hale gelmiştir. Ayrıca, yüksek çözüm kalitesi ve esneklik sunmaları nedeniyle tercih edilmektedir (Kirkpatrick ve ark.,1983; Rashedi ve ark., 2009; Rao, 2016; Mirjalili ve Lewis, 2016).

3.Biyoloji Tabanlı Optimizasyon Algoritmaları

Biyoloji tabanlı optimizasyon algoritmaları, biyolojik süreçlerden, evrimsel prensiplerden veya hayvan ve bitki davranışlarından ilham alınarak tasarlanan meta-sezgisel yöntemlerdir. Bu algoritmalar, doğada gözlemlenen karmaşık sistemleri taklit ederek optimizasyon problemlerine etkili çözümler sunar. Literatür incelendiğinde Biyoloji tabanlı optimizasyon algoritmalarının bir kısmı Şekil 3'te listelenmektedir.

Evrimsel Tabanlı Algoritmalar: Evrimsel tabanlı algoritmalar, doğal seçilim ve genetik varyasyon süreçlerini model alır. Genetik Algoritma (GA)**,** bu yaklaşıma en bilinen örnektir ve çaprazlama, mutasyon gibi genetik işlemleri kullanarak çözümler oluşturur. Diferansiyel Evrim (DE) **ve** Genetik Programlama (GP) ise evrimsel süreçleri temel alan diğer algoritmalardır. Bu algoritmalar genellikle yüksek boyutlu optimizasyon problemlerinde etkili sonuçlar sunar (Holland, 1992).

Hayvan Davranışlarından Esinlenen Algoritmalar*:* Hayvanların doğal davranışlarından ilham alınan algoritmalar, sosyal sürü hareketlerinden bireysel stratejilere kadar birçok özelliği model alır. Örneğin, Karınca Koloni Optimizasyonu (ACO), karıncaların yiyecek arama davranışlarını; Parçacık Sürü Optimizasyonu (PSO) ise kuş sürülerinin kolektif hareketini taklit eder. Bu algoritmalar, yol bulma ve veri kümeleme gibi

problemlerde oldukça etkilidir (Dorigo & Gambardella, 1997; Kennedy & Eberhart, 1995).

Bitki ve Ekosistem Tabanlı Algoritmalar*:* Bitki büyüme süreçleri ve ekosistem içindeki enerji akışı gibi fenomenler, bazı optimizasyon algoritmalarının temelini oluşturur. Yapay Çiçek Tozlaşması (FPA)**,** bitkilerin tozlaşma süreçlerinden esinlenirken**,** Ecosystem-Based Optimization (EBO)**,** enerji akışı ve kaynak paylaşımını model alır. Bu algoritmalar, özellikle sürdürülebilirlik ve çevre ile ilgili optimizasyon problemlerinde kullanılır (Yang, 2012).

Hücresel ve Genetik Sistemlerden Esinlenen Algoritmalar*:* Biyolojinin mikro seviyesinden ilham alan algoritmalar, bağışıklık sistemi veya hücre bölünmesi gibi süreçleri taklit eder. Yapay Bağışıklık Sistemi (AIS)**,** vücut savunma mekanizmalarını model alırken**,** Bakteri Beslenme Optimizasyonu (BFO) bakterilerin besin arama davranışlarını simüle eder. Bu algoritmalar biyoinformatik ve kimyasal süreç optimizasyonunda geniş bir uygulama alanı bulur (Dasgupta, 1999; Passino, 2002).

Evrimsel Tabanlı Algoritmalar

- Genetik Algoritma (GA)
- Diferansiyel Evrim (DE)
- Genetik Programlama (GP)
- Evrim Stratejileri (ES)
- Kültürel Algoritma (CA)

Hayvan Davranışlarından Esinlenen Algoritmalar

- Karınca Koloni Optimizasyonu (ACO)
- Parçacık Sürü Optimizasyonu (PSO)
- Arı Koloni Optimizasyonu (ABC)
- Kurt Sürü Optimizasyonu (GWO)
- Balina Optimizasyon Algoritması (WOA)
- Cırcır Böceği Algoritması (CBA)

Bitki ve Ekosistem Tabanlı Algoritmalar

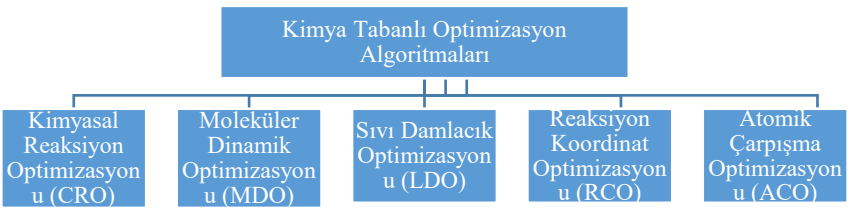
- Yapay Çiçek Tozlaşması (FPA)
- Ekosistem Tabanlı Optimizasyon (EBO)
- Bitki Büyüme Simülasyonu Algoritması (PGSA)

Şekil 3: Biyoloji tabanlı optimizasyon algoritmaları

4.Kimya Tabanlı Optimizasyon Algoritmaları

Kimya tabanlı optimizasyon algoritmaları, kimyasal süreçlerin matematiksel modellerinden esinlenerek geliştirilmiş meta-sezgisel yöntemlerdir. Bu algoritmalar, kimyasal reaksiyonlar, moleküler etkileşimler ve enerji transferleri gibi doğal süreçleri taklit ederek karmaşık optimizasyon problemlerine yenilikçi çözümler sunar. Özellikle, kimyasal reaksiyon optimizasyonu (CRO) gibi algoritmalar, enerji değişimlerini ve reaksiyon yollarını

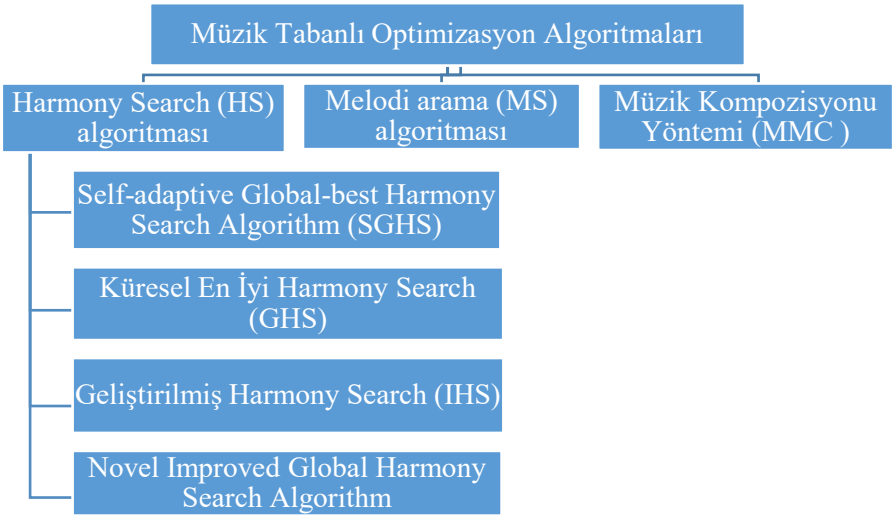
modelleyerek yüksek performanslı optimizasyon sağlar (Lam & Li, 2010). Literatür incelendiğinde Kimya tabanlı optimizasyon algoritmalarının bir kısmı Şekil 4'te listelenmektedir.



Şekil 4: Kimya tabanlı optimizasyon algoritmaları

5.Müzik Tabanlı Optimizasyon Algoritmaları

Müzik tabanlı optimizasyon algoritmaları, müziğin temel bileşenleri olan harmoni, ritim, melodi ve doğaçlamadan esinlenerek geliştirilmiş meta-sezgisel yöntemlerdir. Bu algoritmalar, müzikal uyumu sağlayan süreçleri matematiksel olarak modelleyerek, optimizasyon problemlerini çözmede etkili ve yenilikçi yaklaşımlar sunar. Özellikle, Harmony Search (HS) algoritması, müzikal uyum arama prensiplerinden esinlenerek, geniş bir çözüm uzayında etkili bir şekilde optimal çözümler sunmasıyla öne çıkmaktadır (Geem ve ark., 2001). Literatür incelendiğinde Müzik tabanlı optimizasyon algoritmalarının bir kısmı Şekil 5'de listelenmektedir.



Şekil 5: Müzik tabanlı optimizasyon algoritmaları (Altay ve Alataş, 2020)

6.Sürü Tabanlı Optimizasyon Algoritmaları

Sürü tabanlı optimizasyon algoritmaları, doğadaki hayvan gruplarının toplu hareket, kaynak arama ve iletişim stratejilerinden esinlenerek geliştirilmiştir. Bu algoritmalar, bireylerin (ajanların) kolektif davranışlarını modelleyerek, çözüm uzayında etkili arama yapmayı sağlar. Özellikle, Parçacık Sürü Optimizasyonu (PSO) ve Karınca Koloni Optimizasyonu (ACO) gibi algoritmalar, geniş uygulama alanlarıyla öne çıkmaktadır (Kennedy & Eberhart, 1995; Dorigo & Gambardella, 1997). Literatür incelendiğinde Sürü tabanlı optimizasyon algoritmalarının bir kısmı Şekil 6'da listelenmektedir.

Şekil 6: Sürü tabanlı optimizasyon algoritmaları

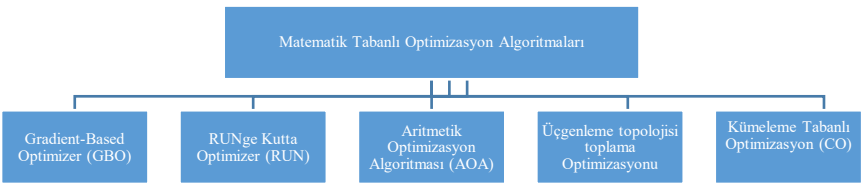
7.Spor Tabanlı Optimizasyon Algoritmaları

Spor tabanlı optimizasyon algoritmaları, sporcuların bireysel performans artırma yöntemleri ve takım dinamikleri gibi spora özgü stratejilerden esinlenerek geliştirilmiş meta-sezgisel yöntemlerdir. Bu algoritmalar, oyun stratejileri ve takım koordinasyonu gibi spor aktivitelerini matematiksel olarak modelleyerek, karmaşık optimizasyon problemlerine etkili çözümler sunar. Özellikle Futbolcu Optimizasyon Algoritması (Soccer Player Optimization, SPO) gibi yaklaşımlar, takım içi iş birliği ve bireysel karar alma süreçlerini simüle etmesiyle dikkat çeker (Basu ve ark., 2014). Literatür incelendiğinde Spor tabanlı optimizasyon algoritmalarının bir kısmı Şekil 7'de listelenmektedir.

Şekil 7: Spor tabanlı optimizasyon algoritmaları (Alataş, 2019)

8.Matematik Tabanlı Optimizasyon Algoritmaları

Matematik tabanlı optimizasyon algoritmaları, matematiksel modeller, teoriler ve kavramlardan esinlenerek geliştirilen meta-sezgisel yöntemlerdir. Bu algoritmalar, doğrusal ve doğrusal olmayan problemlerde matematiksel fonksiyonların özelliklerini kullanarak optimal çözümler sunmayı hedefler. Özellikle Simulated Annealing (SA) ve Gravitational Search Algorithm (GSA) gibi algoritmalar, enerji minimizasyonu ve kütleçekim yasalarına dayalı süreçleri modelleyerek geniş bir uygulama alanına sahiptir (Kirkpatrick ve ark., 1983; Rashedi ve ark., 2009). Literatür incelendiğinde Matematik tabanlı optimizasyon algoritmalarının bir kısmı Şekil 8'de listelenmektedir.



Şekil 8: Matematik tabanlı optimizasyon algoritmaları (Zhao ve ark., 2024)

SONUÇ

Optimizasyon yöntemlerinin günümüz dünyasında önemi oldukça yüksektir. Firmaların maliyetlerini düşürme ve karlarını yükseltme gibi etkisi büyük işler için optimizasyon yöntemleri kullanılmaktadır. Kesin (Tam) ve yaklaşık yöntemler olarak iki ana kategoriye ayrılan optimizasyon yöntemleri ile ilgili birçok

mevcuttur. En çok çalışma ise yaklaşık yöntemler kategorisinde bulunan meta-sezgisel optimizasyon yöntemleri üzerine yapılmaktadır. Çünkü en az maliyetle optimum sonucu veren yöntemlerdir. Bu çalışmada ise meta-sezgisel yöntemlerin sınıflandırması üzerine bir inceleme yapılmaktadır. Birçok alandan ilham alınarak ortaya atılan yeni meta-sezgisel yöntemler, farklı farklı optimizasyon problemlerine uygulanarak başarılı sonuçlar elde edilmektedir. Meta-sezgisel optimizasyon yöntemi önerenler en çok doğadan ilham almaktadırlar. Çünkü doğanın kendisi de optimizasyon yapmaktadır. Doğanın mükemmel tasarımı, meydana gelen olaylar, içinde bulunan canlıların hayatta kalma çabaları gibi birçok yönüyle ilham vermektedir.

KAYNAKLAR

Altunbey, F., & Alataş, B. (2015). Sosyal Ağ Analizi İçin Sosyal Tabanlı Yapay Zekâ Optimizasyon Algoritmalarının İncelenmesi. International Journal of Pure and Applied Sciences, 1(1), 33-52.

Askari, Q., Saeed, M., & Younas, I. (2020). Heap-based optimizer inspired by corporate rank hierarchy for global optimization. *Expert Systems with Applications*, 149, 113702. https://doi.org/10.1016/j.eswa.2020.113702

Basu, S., Saha, S., & Basu, M. (2014). Soccer player optimization: A new meta-heuristic algorithm for global optimization. *Applied Soft Computing, 29*, 475–493. https://doi.org/10.1016/j.asoc.2014.02.035

Dasgupta, D. (1999). Artificial immune systems and their applications. Springer.

Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the traveling salesman problem. Biosystems, 43(2), 73–81. https://doi.org/10.1016/S0303-2647(97)01708-5

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 1*(1), 53–66. https://doi.org/10.1109/4235.585892

Fesanghary, M., Damangir, E., & Ardehali, M. M. (2008). Design optimization of HVAC systems using harmony search algorithm. *Energy and Buildings, 40*(6), 1041–1050. https://doi.org/10.1016/j.enbuild.2007.10.009

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation, 76*(2), 60–68. https://doi.org/10.1177/003754970107600201

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation, 76*(2), 60–68. https://doi.org/10.1177/003754970107600201

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley.

Holland, J. H. (1992). Adaptation in natural and artificial systems. MIT Press.

Juan, A. A., Keenan, P., Martí, R., McGarraghy, S., Panadero, J., Carroll, P., & Oliva, D. (2023). A review of the role of heuristics in stochastic optimisation: From metaheuristics to learnheuristics. *Annals of Operations Research*, *320*(2), 831-861.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459–471. https://doi.org/10.1007/s10898-007-9149-x

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671

Lam, A. Y. S., & Li, V. O. K. (2010). Chemical-reaction-inspired metaheuristic for optimization. *IEEE Transactions on Evolutionary Computation, 14*(3), 381–399. https://doi.org/10.1109/TEVC.2009.2033580

Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software, 95*, 51–67.

Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software, 95*, 51–67.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine, 22(3), 52–67. https://doi.org/10.1109/MCS.2002.1004010

Rajwar, K., Deep, K., ve Das, S. (2023). An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. *Artificial Intelligence Review*, *56*(11), 13187-13257.

Rao, R. V. (2016). *Engineering optimization: Theory and practice* (4th ed.). Wiley.

Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences, 179*(13), 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004

Toaza, B., ve Esztergár-Kiss, D. (2023). A review of metaheuristic algorithms for solving TSP-based scheduling optimization problems. *Applied Soft Computing*, 110908.

Yang, X. S. (2010). Engineering optimization: an introduction with metaheuristic applications. John Wiley & Sons.

Yang, X. S. (2012). Flower pollination algorithm for global optimization. Unconventional Computation and Natural Computation Lecture Notes in Computer Science, 7445, 240–249. https://doi.org/10.1007/978-3-642-32894-7_27

Demirol, D., Oztemiz, F., & Karci, A. "Performance Comparison of Physics Based Meta-Heuristic Optimization Algorithms," *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, Malatya, Turkey, 2018, pp. 1-5, doi: 10.1109/IDAP.2018.8620742.

Altay, E. V., & Alatas, B. (2020). Randomness as source for inspiring solution search methods: Music based approaches. *Physica A: Statistical Mechanics and its Applications*, *537*, 122650.

Alatas, B. (2019). Sports inspired computational intelligence algorithms for global optimization. *Artificial Intelligence Review*, *52*, 1579-1627.

Zhao, S., Zhang, T., Cai, L., & Yang, R. (2024). Triangulation topology aggregation optimizer: A novel mathematics-based meta-heuristic algorithm for continuous optimization and engineering applications. *Expert Systems with Applications*, *238*, 121744.

BÖLÜM IV

Kara Kutudan Açıklanabilir Yapay Zekaya

Musa ATAŞ[1]
Bashar ALHAJAHMAD[2]

Giriş

Yapay zeka (YZ) teknolojileri, son yıllarda büyük bir gelişme göstererek günlük yaşamımızın ve çeşitli endüstrilerin vazgeçilmez bir parçası haline gelmiştir (LeCun ve ark., 2015). Makine öğrenimi ve özellikle derin öğrenme modellerinin artan karmaşıklığı, bu sistemlerin performansını önemli ölçüde artırırken, aynı zamanda kararlarının arkasındaki mantığı anlamayı zorlaştırmıştır (Goodfellow ve ark., 2016; Zhang ve ark., 2018). Bu durum, YZ sistemlerinin güvenilirliği, adilliği ve hesap verebilirliği konusunda ciddi endişelere yol açmıştır (Doshi-Velez & Kim, 2017; Rudin,

[1] Prof. Dr., Siirt Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Siirt/Türkiye, Orcid: 0000-0002-9406-0076, hakmesyo@gmail.com
[2] Dr. Öğr. Üyesi, Siirt Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Siirt/Türkiye, Orcid: 0009-0009-3455-7206, bashar.aptech@gmail.com

2019). Açıklanabilir Yapay Zeka (AYZ) kavramı, bu endişeleri gidermek ve YZ sistemlerinin karar verme süreçlerini insanlar için anlaşılır yapmak amacıyla ortaya çıkmıştır (Adadi & Berrada, 2018; Gunning & Aha, 2019). AYZ, yapay zeka modellerinin nasıl ve neden belirli sonuçlara ulaştığını açıklayabilme yeteneğini öne çıkarır ve son yıllarda araştırmacıların, uygulayıcıların ve politika yapıcıların ilgi odağı haline gelmiştir (Arrieta ve ark., 2020; Miller, 2019).

AYZ'nin önemi, birbirini tamamlayan çeşitli kritik faktörlerden kaynaklanmaktadır. Ribeiro ve arkadaşlarının (2016) çalışmasında vurgulandığı ve Lundberg ve Lee'nin (2017) araştırmalarında doğrulandığı üzere, AYZ öncelikle kullanıcıların yapay zeka sistemlerine olan güvenini artırmada temel bir rol oynamaktadır. Bununla birlikte, yasal uyumluluk açısından, özellikle Avrupa Birliği'nin Genel Veri Koruma Yönetmeliği (GDPR) gibi düzenlemeler, kişiler hakkında otomatik kararlar alan sistemlerin açıklanabilir olmasını zorunlu kılmaktadır (Goodman & Flaxman, 2017; Wachter ve ark., 2017).

AYZ'nin bir diğer kritik rolü, YZ sistemlerinin etik kullanımını sağlamaktır. Doshi-Velez ve Kim (2017) ile Mittelstadt ve arkadaşlarının (2019) çalışmalarında belirtildiği gibi, açıklanabilirlik, yapay zeka sistemlerinin adil ve ayrımcılık yapmayan kararlar almasını sağlamada hayati öneme sahiptir. Molnar (2019) ve Samek ve arkadaşlarının (2017) vurguladığı üzere, açıklanabilir modeller, hataların ve önyargıların daha kolay tespit edilmesini ve düzeltilmesini mümkün kılmaktadır. Bu özellikle modern yapay zeka sistemlerinde kullanılan derin öğrenme modellerinin halen kara kutu sayılan yapay sinir ağlarını kullanması

düşünüldüğünde oldukça önemli olamkla birlikte modelin neden hata yaptığını anlamlandırmayı kritik bir mesele haline getirmektedir.

Son olarak, Lipton'ın (2018) belirttiği gibi, AYZ, yapay zeka alanındaki bilimsel anlayışımızı derinleştirerek daha gelişmiş ve güvenilir sistemlerin oluşturulmasına önemli katkılar sağlamaktadır. Bu bilimsel ilerleme, yapay zeka teknolojilerinin gelecekteki gelişimi için sağlam bir temel oluşturmaktadır.

Bununla birlikte, AYZ uygulamaları çeşitli sektörlerde giderek yaygınlaşmaktadır. Sağlık hizmetlerinde, AYZ teknikleri tıbbi teşhis ve tedavi önerilerinin arkasındaki mantığı açıklamak için kullanılmaktadır (Holzinger ve ark., 2017; Tjoa & Guan, 2020). Finans sektöründe, kredi kararlarının şeffaflığını artırmak ve risk değerlendirmelerini açıklamak için AYZ yöntemleri uygulanmaktadır (Bracke ve ark., 2019). Otonom araçlar alanında, sürüş kararlarının anlaşılabilirliğini sağlamak için AYZ teknikleri geliştirilmektedir (Kim ve ark., 2017; Kim ve ark., 2018). Bununla birlikte, AYZ'nin gelişimi ve uygulanması çeşitli zorluklarla karşı karşıyadır. Karmaşık YZ modellerinin, özellikle derin öğrenme ağlarının, tam olarak açıklanması hala aşılması gereken önemli bir zorluk teşkil etmektedir (Gilpin ve ark., 2018). Ayrıca, açıklamaların insan kullanıcılar tarafından anlaşılabilir ve yorumlanabilir olması gerekmektedir (Miller, 2019). AYZ yöntemlerinin model performansı üzerindeki potansiyel olumsuz etkileri de dikkatle değerlendirilmelidir (Rudin, 2019). Bu çalışmada, AYZ'nin temel prensipleri, mevcut teknikleri, uygulama alanları ve karşılaştığı zorluklar detaylı bir şekilde ilerleyen bölümlerde detaylı olarak ele alınacaktır. İlk olarak, AYZ'nin teorik temellerini ve tarihsel gelişimi

işlenecektir. Ardından, AYZ teknikleri ve yöntemlerini, model-agnostik ve model-spesifik yaklaşımlar çerçevesinde değerlendirilecektir. AYZ'nin çeşitli sektörlerdeki uygulamalarının incelendiği ve karşılaşılan zorlukların tartışıldığı bölümlerden sonra, AYZ'nin geleceği ve potansiyel etkileri yorumlanacaktır.

Bu kitap bölümü, AYZ alanındaki mevcut durumu kapsamlı bir şekilde sunarak, araştırmacılar, uygulayıcılar ve politika yapıcılar için değerli bir kaynak oluşturmayı hedeflemektedir. Açıklanabilir Yapay Zeka sistemlerinin geliştirilmesi, sadece teknolojik bir gereklilik değil, aynı zamanda toplumsal bir sorumluluktur. Bu kitap bölümü, bizi daha şeffaf, güvenilir ve etik yapay zeka sistemlerine doğru götüren adımlar atmaya katkıda bulunacaktır.

Açıklanabilir Yapay Zekanın Teorik Temelleri

Bu bölümde AYZ'nin tarihsel gelişimi, temel prensipleri ile hedefleri ve beklenen olası faydaları üzerinde durulacaktır.

1. AYZ'nin Tarihsel Gelişimi

Açıklanabilir Yapay Zeka kavramı, yapay zeka sistemlerinin karmaşıklığı arttıkça ortaya çıkan bir süreçtir. AYZ'nin kökleri, 1970'lerde geliştirilen uzman sistemlere kadar uzanmaktadır. Bu sistemler, o zamanlarda belirli bir alanda uzman bilgisini kodlayarak karar verme süreçlerini açıklayabiliyordu (Shortliffe & Buchanan, 1975). 1980'ler ve 1990'larda, makine öğrenimi algoritmalarının gelişmesiyle birlikte, bu algoritmaların kararlarını açıklama ihtiyacı da ortaya çıktı. Özellikle karar ağaçları gibi bazı algoritmalar, yapıları gereği daha açıklanabilir modeller sunuyordu (Quinlan, 1986). 2000'li yılların başında, destek vektör makineleri (SVM) ve yapay sinir ağları gibi daha karmaşık modellerin yaygınlaşmasıyla,

bu modellerin kararlarını açıklama zorluğu daha da belirgin hale geldi. Bu dönemde, model yorumlama teknikleri üzerine çalışmalar hız kazandı (Ribeiro ve ark., 2016). 2010'ların ortalarından itibaren, derin öğrenme modellerinin çeşitli alanlarda elde ettiği başarılar, AYZ'ye olan ilgiyi daha da artırdı. Bu dönemde, görüntü sınıflandırma modellerinin kararlarını görselleştirme teknikleri gibi yeni yaklaşımlar ortaya çıktı (Zeiler & Fergus, 2014).

2. AYZ'nin Temel Prensipleri

AYZ'nin temel prensipleri, yapay zeka sistemlerinin karar alma süreçlerini insanlar için anlaşılır kılmayı hedeflemektedir. Doshi-Velez ve Kim'in 2017'deki çalışmasında detaylıca incelenen bu prensipler, birbiriyle yakından ilişkili birkaç temel kavramı kapsamaktadır. Öncelikle, sistemin iç işleyişinin görünür ve anlaşılır olmasını ifade eden şeffaflık prensibi, AYZ'nin temelini oluşturur. Bununla birlikte, modelin kararlarının insan tarafından yorumlanabilir olması ve belirli bir sonuca nasıl ulaştığının açıklanabilmesi de kritik bir öneme haizdir. Bu yorumlanabilirlik ve açıklanabilirlik özellikleri, kullanıcıların yapay zeka sisteminin mantık yürütme sürecini kavramasına imkan tanır. Ayrıca, modelin ürettiği sonuçların doğruluğunun kontrol edilebilmesi ve tutarlı, güvenilir çıktılar üretme kapasitesi de AYZ'nin vazgeçilmez unsurlarındandır.

Söz konusu prensiplere ek olarak, modelin adil olması ve önyargılardan arındırılmış kararlar verebilmesi de üzerinde durulması gereken bir husustur. Ayrıca, sistemin farklı kullanıcı gruplarının ihtiyaçlarına uyarlanabilir olması ve gerektiğinde güncellenebilmesi de modern AYZ uygulamalarında göz önünde bulundurulması gereken faktörlerdendir. Bu prensipler bir bütün

olarak ele alındığında, AYZ'nin sadece teknik bir gereklilik değil, aynı zamanda etik ve sosyal bir sorumluluk olduğu da ortaya çıkmaktadır.

3. AYZ'nin Hedefleri ve Beklenen Faydaları

AYZ'nin temel hedefleri, yapay zeka sistemlerinin toplum tarafından benimsenmesi ve etkin kullanımı açısından kritik öneme sahiptir. Gunning ve Aha'nın 2019'daki çalışmasında vurguladıkarı gibi, öncelikli hedef kullanıcıların yapay zeka sistemlerine olan güvenini artırmaktır. Bu güven ortamının oluşturulması, yapay zeka destekli karar verme süreçlerinin kalitesinin yükseltilmesiyle doğrudan ilişkilidir. Ayrıca, özellikle General Data Protection Regulation (GDPR) gibi düzenlemelere uyum sağlayarak yasal gereklilikleri karşılamak, AYZ'nin önemli hedeflerinden biridir.

Bununla birlikte etik yapay zeka gelişimi kapsamında, adil ve ayrımcılık yapmayan sistemlerin tasarlanması ve uygulanması da büyük önem taşımaktadır. Bu çabalar, aynı zamanda yapay zeka alanında daha derin bir bilimsel anlayış geliştirmeye de katkı sağlamaktadır. Tüm bu hedefler doğrultusunda, AYZ'nin beklenen faydaları arasında daha güvenilir yapay zeka sistemlerinin geliştirilmesi, insan-YZ işbirliğinin güçlendirilmesi ve kullanıcı kabulünün artırılması yer almaktadır. Bu gelişmeler sayesinde, YZ sistemlerinin daha geniş uygulama alanlarında etkin bir şekilde kullanılabilmesi mümkün olmaktadır. Ayrıca, AYZ'nin hedeflerinin gerçekleştirilmesi, yapay zeka teknolojilerinin toplumsal fayda için kullanılmasını ve potansiyel risklerin minimize edilmesini sağlayarak, sürdürülebilir bir yapay zeka ekosisteminin oluşturulmasına katkıda bulunmaktadır.

AYZ Teknikleri ve Yöntemleri

AYZ teknikleri ve yöntemleri, yapay zeka modellerinin kararlarını açıklamak için kullanılan çeşitli yaklaşımları içerir. Bu bölümde, yaygın olarak kullanılan AYZ tekniklerini ve yöntemlerini inceleyeceğiz. Bu teknikleri genel olarak iki ana kategoriye ayırabiliriz: model-agnostik yöntemler ve model-spesifik yöntemler.

1. Model-Agnostik Yöntemler

Model-agnostik yöntemler, herhangi bir makine öğrenimi modeline uygulanabilen ve modelin iç yapısından bağımsız olan tekniklerdir. Bu yöntemler, farklı model türleri için genel bir açıklama çerçevesi sağlar. Bu kategorideki önemli tekniklerden biri, Ribeiro ve arkadaşları (2016) tarafından önerilen LIME (Local Interpretable Model-agnostic Explanations) yöntemidir. LIME, karmaşık modellerin yerel davranışını açıklamak için basit, yorumlanabilir modeller oluşturur. Bu yöntem, bir tahmin için modelin davranışını yerel olarak yaklaşık olarak tahmin eder ve bu yaklaşımı kullanarak açıklamalar üretir. LIME'ın en büyük avantajı, herhangi bir modele uygulanabilmesi ve belirli tahminler için yerel açıklamalar sağlamasıdır. Ancak, bu yerel açıklamalar her zaman global model davranışını yansıtmayabilir ve açıklamaların tutarlılığı seçilen komşuluk boyutuna bağlıdır.

Bir diğer önemli model-agnostik yöntem, Lundberg ve Lee (2017) tarafından geliştirilen SHAP (SHapley Additive exPlanations) yaklaşımıdır. SHAP, oyun teorisinden esinlenen Shapley değerlerini kullanarak her bir özelliğin model çıktısına olan katkısını hesaplar. Bu yöntem, tutarlı ve teorik olarak sağlam bir temel sunar ve hem yerel hem de global açıklamalar sağlayabilir.

Bununla birlikte, SHAP'ın en büyük dezavantajı, özellikle yüksek boyutlu veri setlerinde hesaplama açısından yoğun olabilmesidir. Ayrıca, Shapley değerlerinin yorumlanması bazen sezgisel olmayabilir, bu da yöntemin pratik uygulamalarında zorluklara neden olabilir.

2. Model-Spesifik Yöntemler

Model-spesifik yöntemler, belirli model türlerine özgü açıklama tekniklerdir. Bu yöntemler genellikle modelin iç yapısını ve parametrelerini kullanır. Karar ağaçları ve kural çıkarımı, bu kategorideki önemli yöntemlerden biridir. Karar ağaçları, doğası gereği yorumlanabilir modellerdir ve karmaşık modellerin davranışını açıklamak için vekil (proxy) olarak hizmet edebilirler (Craven & Shavlik, 1995). Bu modeller, doğrudan yorumlanabilir olmaları ve sezgisel, anlaşılması kolay kurallar üretmeleri bakımından avantajlıdır. Ancak özellikle de büyük ağaçlar özelinde, karmaşık ilişkileri yakalamada sınırlı olabilirler. Dikkat mekanizmaları (attention mechanism), özellikle doğal dil işleme ve görüntü analizi alanlarında yaygın olarak kullanılan bir diğer model-spesifik tekniktir. Bu mekanizmalar, modelin belirli giriş bileşenlerine ne kadar "dikkat ettiğini" göstererek açıklamalar sağlar (Vaswani ve ark., 2017). Dikkat mekanizmalarının en büyük avantajı, modelin odaklandığı alanları görselleştirmeyi sağlaması ve sezgisel, anlaşılması kolay açıklamalar sunmasıdır. Bununla birlikte, bu mekanizmalar her zaman gerçek nedensel ilişkileri yansıtmayabilir ve bazı durumlarda yanıltıcı olabilir (Jain & Wallace, 2019). Bu nedenle, dikkat mekanizmalarını kullanırken sonuçları dikkatle yorumlamak önemlidir.

3. Görselleştirme Teknikleri

Görselleştirme tekniklerinin, AYZ'de önemli bir yeri vardır. Bu teknikler, modelin davranışını ve kararlarını görsel olarak temsil ederek daha kolay anlaşılmasını sağlar. Özellik önem grafikleri, bir modelin kararlarında hangi özelliklerin en etkili olduğunu gösterir. Bu grafikler, genellikle çubuk grafikler veya ısı haritaları şeklinde sunulur ve modelin karar verme sürecinde hangi girdilerin daha fazla ağırlığa sahip olduğunu anlamak için kullanılır (Molnar, 2019). Kısmi bağımlılık grafikleri ise bir özelliğin değişiminin model çıktısı üzerindeki etkisini gösterir. Bu grafikler, özelliklerin modelin tahminleri üzerindeki marjinal etkisini anlamak için kullanılır ve model davranışının daha derinlemesine analizini sağlar (Friedman, 2001). Öte yandan, görselleştirme teknikleri, karmaşık yapay zeka modellerinin iç işleyişini anlamada da kullanılabilir. Ataş (2024), derin evrişimli GAN'larda gizli uzay vektörünün hayvan yüzleri üretimindeki etkisini analiz ederek, bu tür modellerin davranışlarını açıklamada görselleştirmenin önemini vurgulamıştır.

AYZ teknikleri ve yöntemleri alanı hızla gelişmektedir ve yeni yaklaşımlar sürekli olarak ortaya çıkmaktadır. Bu bölümde sunulan teknikler, alanın temelini oluşturan ve yaygın olarak kullanılan yöntemlerdir. Her bir tekniğin kendi güçlü ve zayıf yönleri vardır ve belirli bir uygulama için en uygun tekniğin seçimi, problemin doğasına, veri setinin özelliklerine ve açıklama ihtiyaçlarına bağlıdır. Araştırmacılar ve uygulayıcılar, bu teknikleri kullanırken ve yeni teknikler geliştirirken, açıklamaların doğruluğu, tutarlılığı ve anlaşılabilirliği arasında bir denge kurmak zorundadırlar. Gelecek bölümlerde, bu tekniklerin çeşitli uygulama alanlarında nasıl

kullanıldığını ve pratik zorluklarla nasıl başa çıkıldığını inceleyeceğiz.

AYZ’nin Uygulama Alanları

Açıklanabilir Yapay Zeka (AYZ), çeşitli sektörlerde giderek daha fazla önem kazanmaktadır. Bu bölümde, AYZ'nin başlıca uygulama alanlarını inceleyeceğiz ve her bir alanda AYZ'nin nasıl kullanıldığını, sağladığı faydaları ve karşılaşılan zorlukları ele alacağız.

1. Sağlık Sektöründe AYZ

Sağlık sektörü, AYZ'nin en kritik uygulama alanlarından biridir. Yapay zeka modelleri, hastalık teşhisi, tedavi planlaması ve ilaç keşfi gibi alanlarda giderek daha fazla kullanılmaktadır. Ancak, sağlık gibi hayati önem taşıyan bir alanda, yapay zeka sistemlerinin kararlarının anlaşılabilir ve açıklanabilir olması büyük önem taşır. AYZ teknikleri, doktorların ve hastaların yapay zeka destekli kararları daha iyi anlamalarına ve değerlendirmelerine olanak tanır. Örneğin, Ribeiro ve arkadaşları (2016) tarafından geliştirilen LIME tekniği, cilt kanseri teşhisinde kullanılan derin öğrenme modellerinin kararlarını açıklamak için kullanılmıştır. Bu uygulama, modelin hangi görüntü özelliklerine dayanarak bir lezyonu kötü huylu olarak sınıflandırdığını göstermiştir. Benzer şekilde, Lundberg ve Lee (2017) tarafından önerilen SHAP yöntemi, hastane yatış süresi tahmininde kullanılan modellerin kararlarını açıklamak için uygulanmıştır. Diğer yandan, AYZ teknikleri, tıbbi görüntü analizinde de önemli rol oynamaktadır. Örneğin, Ataş (2023a) diyabetik retinopati görüntülerinin sentezinde derin evrişimli ve en küçük kareler GAN'larını karşılaştırarak, bu modellerin

performansını ve potansiyel AYZ uygulamalarını incelemiştir. Bu çalışmalar, AYZ'nin sağlık profesyonellerine YZ sistemlerinin kararlarını değerlendirme ve doğrulama imkanı sunduğunu göstermiştir. Bununla birlikte, sağlık sektöründe AYZ uygulamaları bazı zorluklarla da karşılaşmaktadır. Örneğin, tıbbi verilerin gizliliği ve hassasiyeti, AYZ tekniklerinin uygulanmasında dikkatli bir yaklaşım gerektirmektedir. Ayrıca, tıbbi kararların karmaşıklığı, bazen basit açıklamaların yetersiz kalmasına neden olabilmektedir. Bu nedenle, sağlık alanında AYZ çalışmaları, hem teknik hem de etik açıdan sürekli geliştirilmeye ihtiyaç duymaktadır.

2. Finans ve Bankacılıkta AYZ

Finans ve bankacılık sektörü, AYZ'nin bir diğer önemli uygulama alanıdır. Kredi skorlaması, dolandırıcılık tespiti ve risk değerlendirmesi gibi kritik finansal kararlar giderek daha fazla yapay zeka sistemlerine dayanmaktadır. Ancak, finansal kararların şeffaflığı ve açıklanabilirliği, yasal düzenlemeler ve müşteri güveni açısından son derece önemlidir. AYZ teknikleri, finansal modellerin kararlarını açıklamak için yaygın olarak kullanılmaktadır. Örneğin, Bracke ve arkadaşları (2019) tarafından yapılan bir çalışmada, makine öğrenimi modellerinin kredi risk değerlendirmesindeki kararlarını açıklamak için çeşitli AYZ teknikleri kullanılmıştır. Bu çalışma, AYZ'nin finansal modellerin şeffaflığını artırmada ve düzenleyici gereksinimleri karşılamada nasıl yardımcı olabileceğini göstermiştir. Finans sektöründe AYZ'nin bir diğer önemli uygulaması, anomali ve dolandırıcılık tespitinde görülmektedir. Karmaşık modeller, olağan dışı işlemleri tespit etmekte oldukça başarılıdır, ancak bu tespitlerin nedenlerinin açıklanması gerekmektedir. AYZ teknikleri, şüpheli işlemlerin neden

işaretlendiğini açıklayarak, hem müşteri hizmetlerinin kalitesini artırmakta hem de yanlış pozitifleri azaltmada yardımcı olmaktadır. Ancak, finans sektöründe AYZ uygulamaları da bazı zorluklarla karşılaşmaktadır. Özellikle, finansal verilerin gizliliği ve modellerin ticari sırları, açıklamaların detay seviyesini sınırlayabilmektedir. Ayrıca, finansal sistemlerin karmaşıklığı, bazen AYZ tekniklerinin ürettiği açıklamaların tam olarak anlaşılmasını zorlaştırabilmektedir.

3. Otonom Araçlarda AYZ

Otonom araçlar, AYZ'nin kritik önem taşıdığı bir diğer alandır. Sürücüsüz araçların karar verme süreçlerinin anlaşılabilir ve açıklanabilir olması, hem güvenlik hem de yasal sorumluluk açısından büyük önem taşımaktadır. AYZ teknikleri, otonom araçların sürüş kararlarını açıklamak için çeşitli şekillerde kullanılmaktadır. Örneğin, Kim ve arkadaşları (2018) tarafından yapılan bir çalışmada, görsel dikkat mekanizmaları kullanılarak otonom araçların hangi görsel unsurlara dayanarak sürüş kararları aldığı açıklanmıştır. Bu tür açıklamalar, araç davranışının anlaşılmasını kolaylaştırmakta ve güvenlik değerlendirmelerinde kritik rol oynamaktadır. Otonom araçlarda AYZ'nin bir diğer önemli uygulaması, etik karar verme süreçlerinin açıklanmasıdır. Örneğin, bir kaza durumunda aracın nasıl karar verdiğini açıklamak, hem yasal hem de etik açıdan büyük önem taşımaktadır. AYZ teknikleri, bu tür karmaşık etik kararların arkasındaki mantığı açıklamada yardımcı olmaktadır. Ancak, otonom araçlarda AYZ uygulamaları bazı zorluklarla da karşı karşıya kalmaktadır. Örnek vermek gerekirse, bazen gerçek zamanlı karar verme süreçlerinin açıklanması teknik açıdan zorlayıcı olabilmektedir. Ayrıca, açıklamaların sürücüler ve yayalar tarafından hızlı ve doğru bir

şekilde anlaşılabilir olması gerekmektedir. Bu nedenle, otonom araçlarda AYZ çalışmaları, hem teknik performans hem de insan-makine etkileşimi açısından sürekli geliştirilmeye ihtiyaç duymaktadır.

AYZ'nin uygulama alanları bu üç sektörle sınırlı değildir. Eğitim, hukuk, savunma ve daha birçok alanda AYZ teknikleri giderek daha fazla önem kazanmktadır. Her bir uygulama alanı, kendine özgü fırsatlar ve zorluklar sunmaktadır. AYZ'nin başarılı bir şekilde uygulanması, hem teknik hem de alana-özgü (domain-spesific) bilgiyi gerektirmektedir. Gelecek bölümlerde, AYZ'nin uygulanmasında karşılaşılan genel zorlukları ve bu zorlukların üstesinden gelmek için geliştirilen stratejileri ele alacağız.

AYZ'de Karşılaşılan Zorluklar

Açıklanabilir Yapay Zeka (AYZ) alanı, yapay zeka sistemlerinin şeffaflığını ve anlaşılabilirliğini artırmada önemli ilerlemeler kaydetmiş olsa da, hala birçok zorluk ve sınırlamayla karşı karşıyadır. Bu bölümde, AYZ'nin karşılaştığı başlıca zorlukları ve sınırlamaları inceleyeceğiz.

1. Teknik Zorluklar

AYZ'nin karşılaştığı en önemli teknik zorluklardan biri, karmaşık yapay zeka modellerinin, özellikle derin öğrenme ağlarının, iç işleyişlerini açıklamaya çalışmaktır. Bu modeller milyonlarca parametre içerebilir ve karar verme süreçleri doğrusal olmayan karmaşık etkileşimlere ve zamana bağlı arttırımlı hesaplamalara dayanabilir. Bu karmaşıklık, anlamlı ve insan tarafından anlaşılabilir açıklamalar üretmeyi zorlaştırır. Örneğin, Zhang ve arkadaşları (2018) tarafından yapılan bir çalışma, derin

sinir ağlarının iç temsillerini yorumlamanın zorluklarını ortaya koymuştur. Araştırmacılar, konvolüsyonel sinir ağlarının (CNN) farklı katmanlarındaki nöronların aktivasyonlarını incelemiş ve bu aktivasyonların insan tarafından anlaşılabilir kavramlarla nasıl ilişkilendirilebileceğini araştırmışlardır. Ancak, derin katmanlardaki temsillerin özelliklede spatial alnadan uzaklaşıldığı ve daha soyut alanlara kaydığı düşünüldüğünde genellikle yorumlanmasının oldukça zor olduğunu tespit etmişlerdir. Bir diğer teknik zorluk, AYZ tekniklerinin model performansı üzerindeki potansiyel olumsuz etkisidir. Bazı durumlarda, bir modeli daha açıklanabilir hale getirmek için yapılan değişiklikler, modelin tahmin doğruluğunu düşürebilir. Bu, özellikle yüksek riskli kararlar alan sistemlerde önemli bir ikilem meydana getirir. Rudin (2019), bu sorunu ele almış ve bazı durumlarda karmaşık kara kutu modeller yerine doğası gereği yorumlanabilir modeller kullanmanın daha uygun olabileceğini savunmuştur. Bununla birlikte, AYZ tekniklerinin performansını değerlendirmek de önemli bir zorluktur. Ataş (2023b), yüksek çözünürlüklü uydu görüntülerinden bina segmentasyonunda Jaccard-Dice katsayısının performansını değerlendirmiştir. Bu tür çalışmalar, yapay zeka yöntemlerinin doğruluğunu ve güvenilirliğini ölçmek için benzer metriklerin kullanılabileceğini bize açıkça göstermektedir.

2. Etik ve Yasal Zorluklar

AYZ, etik ve yasal açıdan da çeşitli zorluklarla karşı karşıyadır. Bunlardan biri, açıklamaların yeterli detay seviyesini belirlemektir. Çok az açıklama, sistemin şeffaflığını sağlamazken, çok fazla açıklama ise kullanıcıya gereksiz bilgi sunarak sistemin güvenliğini tehlikeye atabilir. Avrupa Birliği'nin Genel Veri Koruma

Yönetmeliği (GDPR) gibi yasal düzenlemeler, otomatik karar verme sistemlerinin "açıklanma hakkı" sunmasını gerektirir. Ancak, Wachter ve arkadaşları (2017) tarafından yapılan bir analiz, bu hakkın pratikte uygulanmasının zorluklarını ortaya koymuştur. Araştırmacılar, mevcut AYZ tekniklerinin her zaman yasal gereksinimleri karşılamaya yetmeyebileceğini ve bu alanda daha fazla çalışma yapılması gerektiğini vurgulamışlardır. Bir diğer etik zorluk, AYZ tekniklerinin potansiyel olarak kötüye kullanılmasıdır. Açıklamalar, kötü niyetli aktörler tarafından sistemin zayıf noktalarını bulmak ve istismar etmek için kullanılabilir. Bu nedenle, açıklanabilirlik ile sistem güvenliği arasında bir dengenin kurulması kaçınılmazdır.

3. Pratik Uygulamalarda Karşılaşılan Zorluklar

AYZ'nin pratik uygulamalarında da çeşitli zorluklarla karşılaşılmaktadır. Bunlardan biri, farklı kullanıcı gruplarının farklı açıklama düzeylerine olan gereksinimleridir. Örneğin, bir yapay zeka sisteminin geliştiricisi, bir son kullanıcı ve bir düzenleyici kurum temsilcisi, aynı sistemden farklı türde ve detayda açıklamalar bekleyebilir. Miller (2019) tarafından yapılan kapsamlı bir inceleme, insanların açıklamaları nasıl algıladığı ve yorumladığı konusunda önemli içgörüler sunmuştur. Çalışma, etkili AYZ sistemlerinin sadece teknik doğruluk değil, aynı zamanda bilişsel psikoloji ve sosyal bilimlerden gelen anlayışları da dikkate alması gerektiğini vurgulamıştır. Bir diğer pratik zorluk, AYZ tekniklerinin gerçek zamanlı sistemlerde uygulanmasıdır. Özellikle hızlı karar verme gerektiren uygulamalarda, açıklamaların zamanında üretilmesi ve sunulması güç olabilir. Bu, özellikle otonom araçlar veya gerçek zamanlı finans uygulamaları gibi alanlarda önemli bir sorundur.

Sonuç

Açıklanabilir Yapay Zeka (AYZ), yapay zeka sistemlerinin giderek karmaşıklaşan ve yaygınlaşan dünyasında kritik bir araştırma alanı haline gelmiştir. Bu bölümde incelediğimiz gibi, AYZ'nin önemi, yapay zeka sistemlerinin karar verme süreçlerini şeffaf ve anlaşılır yapma ihtiyacından kaynaklanmaktadır. AYZ, sadece teknik bir gereklilik değil, aynı zamanda etik, yasal ve toplumsal bir zorunluluktur. AYZ'nin teorik temelleri, yapay zeka sistemlerinin açıklanabilirliğini sağlamak için sağlam bir çerçeve sunmaktadır. Bu temeller üzerine inşa edilen çeşitli teknikler ve yöntemler, karmaşık yapay zeka modellerinin kararlarını yorumlamak ve açıklamak için kullanılmaktadır. Model-agnostik yöntemler, herhangi bir yapay zeka modeline uygulanabilirken, model-spesifik yöntemler belirli model türlerine özel çözümler sunmaktadır. Bu tekniklerin her birinin kendi ait güçlü ve zayıf yönleri vardır ve uygun tekniğin seçimi, problemin yapısına ve ondan istene açıklama düzeyi gibi isterlere bağlıdır.

AYZ'nin uygulama alanları, sağlık, finans, otonom araçlar ve daha birçok sektörü kapsamaktadır. Her bir alanda, AYZ teknikleri yapay zeka sistemlerinin kararlarını açıklamak, güven oluşturmak ve yasal gereklilikleri karşılamak için kullanılmaktadır. Örneğin, sağlık sektöründe AYZ, tıbbi teşhis ve tedavi planlamasında kullanılan modellerin kararlarını açıklayarak, hem doktorlara hem de hastalara değerli içgörüler sunmaktadır. Finans sektöründe, AYZ teknikleri kredi kararlarının şeffaflığını artırmakta ve risk değerlendirmelerini açıklamaktadır. Otonom araçlarda ise, AYZ sürüş kararlarının anlaşılmasını sağlayarak güvenlik ve yasal sorumluluk konularında kritik bir rol oynamaktadır.

Ancak, AYZ alanı hala birçok zorluk ve sınırlamayla karşı karşıyadır. Teknik zorluklar, özellikle derin öğrenme gibi karmaşık modellerin açıklanmasında ortaya çıkmaktadır. Etik ve yasal zorluklar, açıklamaların yeterli detay seviyesini belirleme ve kötüye kullanımı önleme gibi konuları içermektedir. Pratik uygulama zorlukları ise, farklı kullanıcı gruplarının ihtiyaçlarını karşılama ve gerçek zamanlı sistemlerde AYZ tekniklerini uygulama gibi konuları kapsamaktadır. Bu zorluklar, AYZ alanında gelecekteki araştırmaların yönünü şekillendirmektedir. Daha etkili ve verimli açıklama teknikleri geliştirmek, farklı kullanıcı ihtiyaçlarına uyarlanabilir açıklama sistemleri tasarlamak ve AYZ'nin etik ve yasal etkilerini daha iyi anlamak, gelecekteki araştırmaların odak noktaları olacaktır. Özellikle, AYZ tekniklerinin standardizasyonu ve değerlendirilmesi için daha iyi yöntemler geliştirmek önemli bir araştırma alanıdır.

AYZ'nin geleceği, yapay zeka sistemlerinin toplum tarafından daha iyi anlaşılmasını ve kabul görmesini sağlama potansiyeline sahiptir. Doshi-Velez ve Kim'in (2017) önerdiği gibi, AYZ alanının ilerlemesi için disiplinler arası işbirliği kritik öneme sahiptir. Bilgisayar bilimi, kognitif psikoloji, sosyal bilimler ve hukuk gibi farklı alanlardan gelen bilgilerin entegrasyonu, daha etkili, açıklanabilir ve anlamlı AYZ sistemlerinin geliştirilmesine yol açacaktır. Sonuç olarak, AYZ, yapay zeka sistemlerinin güvenilirliğini, şeffaflığını ve hesap verebilirliğini artırma potansiyeli ile gelecekteki yapay zeka araştırmalarının ve uygulamalarının ayrılmaz bir parçası haline gelmiştir. AYZ'nin karşılaştığı zorlukların üstesinden gelmek, sadece daha iyi yapay zeka sistemleri geliştirmeyi değil, aynı zamanda bu sistemlerin

toplum tarafından daha iyi anlaşılmasını ve kabul görmesini de sağlayacaktır. Bu nedenle, AYZ alanındaki araştırmalar ve uygulamalar, güvenilir ve etik yapay zeka sistemlerinin geliştirilmesi ve yaygınlaştırılması için kritik öneme sahiptir.

Gelecekte, AYZ'nin sadece mevcut yapay zeka sistemlerini açıklamakla kalmayıp, aynı zamanda daha yorumlanabilir ve şeffaf yapay zeka sistemlerinin tasarımına rehberlik etmesi beklenmektedir. Bu, yapay zekanın insan zekasını tamamlayıcı bir araç olarak daha etkili kullanımına yol açabilir ve insan-YZ işbirliğinin yeni formlarının önünü açabilir. AYZ, yapay zeka teknolojilerinin sorumlu ve etik gelişimi için bir katalizör görevi görerek, toplumun yapay zekaya olan güvenini artırabilir ve yapay zekanın potansiyelinin tam olarak gerçekleştirilmesine katkıda bulunabilir.

Kaynaklar

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access, 6, 52138-52160.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82-115.

Ataş, İ. (2024). The Effect of Latent Space Vector on Generating Animal Faces in Deep Convolutional GAN: An Analysis. DUJE, 15(1), 99-106. doi: 10.24012/dumf.1393797

Ataş, İ. (2023a). Comparison of deep convolution and least squares GANs for diabetic retinopathy image synthesis. Neural Computing and Applications, 35, 14431-14448. https://doi.org/10.1007/s00521-023-08482-4

Ataş, İ. (2023b). Performance Evaluation of Jaccard-Dice Coefficient on Building Segmentation from High Resolution Satellite Images. Balkan Journal of Electrical and Computer Engineering, 11(1), 100-106. https://doi.org/10.17694/bajece.1212563

Bracke, P., Datta, A., Jung, C., & Sen, S. (2019). Machine learning explainability in finance: An application to default risk analysis. Bank of England Staff Working Paper No. 816.

Craven, M., & Shavlik, J. W. (1995). Extracting tree-structured representations of trained networks. Advances in Neural Information Processing Systems, 8, 24-30.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of Statistics, 29(5), 1189-1232.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA) (pp. 80-89).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

Goodman, B., & Flaxman, S. (2017). European Union regulations on algorithmic decision-making and a 'right to explanation'. AI Magazine, 38(3), 50-57.

Gunning, D., & Aha, D. W. (2019). DARPA's explainable artificial intelligence program. AI Magazine, 40(2), 44-58.

Holzinger, A., Biemann, C., Pattichis, C. S., & Kell, D. B. (2017). What do we need to build explainable AI systems for the medical domain?. arXiv preprint arXiv:1712.09923.

Jain, S., & Wallace, B. C. (2019). Attention is not explanation. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1, 3543-3556.

Kim, J., Rohrbach, A., Darrell, T., Canny, J., & Akata, Z. (2018). Textual explanations for self-driving vehicles. In

Proceedings of the European Conference on Computer Vision (ECCV) (pp. 563-578).

Kim, J., & Canny, J. (2018). Explainable deep driving by visualizing causal attention. Explainable and Interpretable Models in Computer Vision and Machine Learning, 173-193.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue, 16(3), 31-57.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems (pp. 4765-4774).

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, 267, 1-38.

Mittelstadt, B., Russell, C., & Wachter, S. (2019). Explaining explanations in AI. In Proceedings of the Conference on Fairness, Accountability, and Transparency (pp. 279-288).

Molnar, C. (2019). Interpretable machine learning. A Guide for Making Black Box Models Explainable. https://christophm.github.io/interpretable-ml-book/

Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81-106.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). 'Why should I trust you?' Explaining the predictions of any classifier. In

Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135-1144).

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5), 206-215.

Samek, W., Wiegand, T., & Müller, K. R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv preprint arXiv:1708.08296.

Shortliffe, E. H., & Buchanan, B. G. (1975). A model of inexact reasoning in medicine. Mathematical Biosciences, 23(3-4), 351-379.

Tjoa, E., & Guan, C. (2020). A survey on explainable artificial intelligence (XAI): Toward medical XAI. IEEE Transactions on Neural Networks and Learning Systems.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998-6008.

Wachter, S., Mittelstadt, B., & Floridi, L. (2017). Why a right to explanation of automated decision-making does not exist in the general data protection regulation. International Data Privacy Law, 7(2), 76-99.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In European Conference on Computer Vision (pp. 818-833).

Zhang, Q., Cao, R., Shi, F., Wu, Y. N., & Zhu, S. C. (2018). Interpreting CNN knowledge via an explanatory graph. In Proceedings of the AAAI Conference on Artificial Intelligence, 32(1).

BÖLÜM V

Parçacık Sürü Optimizayon (PSO) Algoritması

UğurTALAŞ

Giriş

Optimizasyon, bir problemin sonucuna ulaşırken yapılan iyileştirme sürecidir. Bu süreci belirli kurallara bağlı algoritma ile sistemli bir şekilde yapılmasına optimizasyon algoritması denilmektedir. Sezgisel optimizasyon algoritmaları genellikle doğadan ilam alınarak geliştirilmiş ve optimizasyon problemlerini çözmeyi hedefleyen yöntemlerdir(Beheshti & Shamsuddin, 2013). Literatürde geliştirilmiş birçok algoritma bulunmaktadır (Kokash, 2005). Çözüme gidiş yolları, gelişim yapıları, hedefledikleri problem tipleri gibi birçok farklı yönden gruplanabilirler. Sürü zekâsı, sürülerin davranışlarını andıran, genellikle kuşlar, balıklar, arılar karıncalar gibi doğal yaşamdaki sürülerin yemek bulma, düşmandan kaçma göç etme gibi doğal problemlerini çözme yeteneklerindeki toplu hareketleri, bu tür algoritmalara ilham kaynağı olmuştur.

Parçacık sürü optimizasyon algoritması da sürü zekasına sahip bir sezgisel optimizasyon algoritmasıdır.

Parçacık Sürü Optimizasyon Algoritması(PSO), doğadan esinlenen ve sürü zekasına sahip sezgisel bir optimizasyon algoritmasıdır. 1995 yılında Kennedy ve Eberhart tarafından, kuş sürülerinin hareketlerinden ilham alınarak geliştirilmiştir(Kennedy & Eberhart, 1995). PSO, çözüm uzayında en uygun çözümü bulmak için bir sürü parçacığın işbirliği içinde matematiksel bir modele dayanarak hareket ettiği bir yöntemdir (Kennedy & Eberhart, t.y.). Problemi çözerken parçacıklar olarak adlandırılan aday çözümleri içeren bir popülasyona sahiptir ve bu parçacıkları kısıtları belirlenen arama uzayında temel basit bir matematiksel formüle göre aday çözümü temsil eden parçacığın konumu ve hızı üzerinden hareket ederek çözüme ulaşır. Popülasyondaki her bir parçacığın hareketi hem kendisinin en iyi olduğu pozisyona hem de sürüdeki elde edilmiş en iyi çözümün pozisyonuna doğru hareketle istenilen çözüme ulaşmaya çalışır(Yılmaz vd., 2022). Sürü hareketi hem bireysel hem de genelde elde edilen uygun çözümlere yaklaşma eğilimdedir ve bu çözümlerden etkilenerek yoluna devam eder. Çok parametreli ve çok değişkenli optimizasyon problemlerinde beklenen sonuca hızlı ulaşması, az parametre gerektirmesi ve yerel optimumlara takılma riskinin az olması sebebiyle literatürde tercih edilen bir algoritma olmuştur. Literatürde sadece benchmark çalışmalarında değil bir çok mühendislik probleminde de yaygın olarak kullanılmıştır (Marini & Walczak, 2015; Özmen vd., 2018).

Dilber Çetintaş ve Taner Tuncer, otizm hastalığında göz hareketlerinden otizm spektrum bozukluğu olan ve otizm spektrum bozukluğu olmayan hastaların tespit için bir çalışma

gerçekleştirilmişlerdir. Çalışmalarında Long Sort Term Memory (LSTM) ve PSO algoritmalarını kullanarak hibrit bir yapı oluşturmuşlardır. Göz hareketlerini izleyip bir dizi haline getirerek LSTM ağında işlemişlerdir. LSTM içerinde pencere boyutunun seçimi başarımı etkileyen önemli bir faktör olduğu için pencere boyutunun seçimi PSO algoritması ile optimize edilmiştir. Hibrit yapının yapılan testeler sonucunda %98,64'lük bir başarım elde ettiği görülmüştür. Otizm tanısı ve tedavisi için önemli bir çalışma gerçekleştirilmiştir(Çetintaş vd., 2023).

H.Lale Zeynelgil yaptığı bir çalışmada, elektrik dağıtım sistemlerinin ağır ve hafif yüklenmesindeki geçişlerden kaynakları kayıpları minimuma indirmek için gün içinde optimal dağıtımının sağlanmasına yönelik PSO algoritmasını kullanmıştır. Yapılan çalışma ile literatürdeki genetik algoritma, çekirge algoritması, harmani algoritması gibi farklı algoritmalar ile karşılaştırılmıştır. Çalışmada önerilen ikili parçacık sürü algoritması İPSO, klasik PSO'ya göre daha iyi yakınsama gösterdiği ifade edilmiştir(Lale Zeynelgil vd., 2023).

Mehmet Kevser DERDİMAN "Ayrık PSO Algoritması İle Konsol Döşemelerin Güvenilirlik Tabanlı Kesit Optimizasyonu" isimli çalışmasında konsol döşemelerinde, döşemenin sehim sınırlarını aşmaması TS500 taşıma gücü kriterini sağlamak için optimizasyon yapmıştır. Yapılan çalışma TS500 sınır şartları ile TS500 taşıma gücü yöntemi ve deplasman kriterleri dikkate alınarak farklı beton dayanımları altında çözüm uzayı içinden optimal çözüme ulaşılmıştır. Çalışmada PSO optimizasyon algoritması kullanılmıştır(Derdiman, 2022).

Aziz ULUSOY ve Mahit GÜNEŞ yaptıkları çalışmada mobil robot kolunun stabilizasyonu için PSO optimizasyon algoritmasını kullanmışlardır. Kullanılan robot kolunda PID parametreleri a gerçek zamanlı alınan verilerin optimize ederek motorların yönetimi optimize edilmiştir. Yapılan çalışma klasik PID kontrolüne göre daha az hata yapığı ve istenilen referans konumunu daha kısa sürede ulaştığı ifade edilmiştir (Ulusoy & Güneş, 2019).

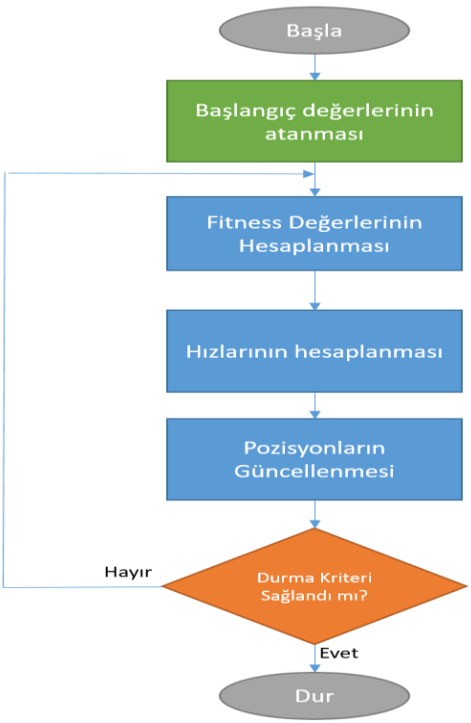
A. Kaveh ve A. Nasrollahi yaptıkları çalışmalarında PSO ve Harmony Search (HS) optimizasyon algoritmalarını birleştirerek iki algoritmanın da keşif ve faydalanmadaki dengesini arttırmayı hedeflemiştir. Yaptıkları çalışmayı yay tasarımı problemi, basınçlı kap problemi gibi farklı mühendislik problemlerine uygulayarak test etmişlerdir(Kaveh & Nasrollahi, 2013).

PSO algoritması literatürdeki çalışmalarda bir çok farklı problemin çözümüne fayda sağlamıştır. Özellikle basit matematiksel yapısı ve genel çalışma yapısının sadeliği farklı problemlere uygulanabilirliğini arttırmaktadır. Parametre ayarlamasının da oldukça basit olan bu algoritma bir sonraki bölümde çalışma mekanizmaları ve matematiksel modeli incelenip, devam eden bölümde X^2 ve Rastrigin örnek problemleri için her bir iterasyondaki elde edilen ölçümler hem de grafik üzerinde global optimuma yakınsaması incelenmiştir.

Parçacık Sürü Optimizasyon Algoritması (PSO)

PSO algoritması ilk olarak 1995 yılında ortaya atılmıştır (Bülbül, 2023) ve o günden günümüze sezgisel optimizasyon algoritmaları arasında en yaygın kullanılanlardan bir tanesi olmuştur (Marini & Walczak, 2015b). En önemli özelliklerinden biri sade ve

basit bir matematiksel alt yapıya dayanmasıdır. PSO temel olarak parçacık hız hesaplaması ve parçacık pozisyon güncellemesi olarak iki farklı matematiksel formül ile güncelleme yapmaktadır (Yi, 2016). Sürü zekasına sahip PSO algoritmasının akış diyagramı aşağıdaki Şekil 1'de verilmiştir.



Şekil 1: PSO akış diyagramı

Yukarıdaki Şekil 1'deki akış diyagramında görüldüğü üzere PSO algoritması rastgele bir sürü ile başlatılır. İlk olarak her bir parçacığın maliyeti hesaplanır. Elde edilen fitness değerlerine göre Pbest ve Gbest güncellemesi yapılır. Sonrasında iteratif bir şekilde parçacıklarının hızları hesaplanır. Bu hızlar kullanılarak bir sonraki

pozisyonu belirlenir ve durma kontrolü yapılır. Durma kontrolü sonrasında gerekli iterasyon tekrarlanır veya algoritma durdurulur. Bu çalışma akışı aşağıda adımlar halinde detaylı olarak verilmiştir.

1. Adım: Parçacıkların Başlatılması

Algoritma uygulayıcı tarafından belirlenen parametrelere göre istenilen sayıda parçacıktan oluşan bir sürü oluşturulur. Oluşturulan sürüdeki her bir parçacık için rastgele pozisyonlar atanır. Pozisyonlar atanırken eğer çözüm kümesi için belirlenen bir kısıt varsa rastgele değerler ilgili çözüm aralığına uygun şekilde rastgele atama yapılır.

2. Adım: Fitness Değerlendirmesi

Popülasyonu oluşturan her bir parçacık problem olarak tanımlanan amaç fonksiyonuna gönderilir ve maliyeti hesaplanır. Amaç fonksiyonu uygulanan probleme göre değişiklik gösterir.

3. Adım: En İyi Konumların Güncellenmesi

Algoritma içerisinde her bir parçacık için o parçacığın en iyi konumu tutulur bu değere Pbest denir. Eğer bir parçacığın mevcut konumu, daha önce bulduğu en iyi çözümden iyiyse, o zaman parçacığın Pbest değeri güncellenir. Algoritma içinde sürünün geneli içinde Gbest adında sürünün en uygun fitness değerine sahip parçacığın pozisyonu tutulur. Eğer bir parçacığın konumu, sürünün mevcut en iyi çözümünden iyiyse, o zaman Gbest değeri güncellenir.

4. Adım: Hızın Güncellenmesi

Bir parçacığın pozisyonunun güncellenmesi için algoritmanın ilk mekanizması olan hız hesaplaması yapılır. Hız hesaplaması için aşağıdaki Eşitlik 1 kullanılır.

$$V_i(t+1) = w * V_i(t) + C_1 * r_1 * \left(P_{best} - X_i(t) \right)$$

$$+C_2 * r_2 * (G_{best} - X_i(t)) \qquad (1)$$

$V_i(t+1)$: Parçacığın yeni hızı,

w : Eylemsizlik katsayısı,

$V_i(t)$: Parçacığın mevcut hızı,

$X_i(t)$: Parçacığın mevcut pozisyonu,

C_1 , C_2 : Öğrenme katsayılar,

r_1 , r_2 :[0 -1] aralığında seçilen rastgele değerler,

P_{best} : Parçacığın şu ana kadar en iyi fitness değeri (Yerel en iyi çözüm)

G_{best} : Sürünün şu ana kadar en iyi fitness değeri (Genel en iyi çözüm)

Yukarıdaki Eşitlik 1'de verilen formül ile sürü içerisindeki tüm parçacıklar için hız hesaplaması yapılır ve hesaplanan hızlar hafızada tutulup bir sonraki adımda pozisyonların güncellemesi için kullanılır.

5.Adım: Konumun Güncellenmesi

Hızları hesaplanan parçacıkların bir sonraki konumlarının belirlendiği adımdır. Konum güncelleme mekanizması aşağıdaki Eşitlik 2'de verilen formül ile yapılır.

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (2)$$

$X_i(t+1)$: Parçacığın yeni pozisyonu,
$V_i(t+1)$: Parçacığın yeni hızı,

$X_i(t)$: Parçacığın mevcut pozisyonu,

Yukarıdaki Eşitlik 2'de verilen formülde parçacığın bir önceki iterasyondaki hızı ile ilgili iterasyondaki hızı toplanarak parçacığın yeni pozisyonu hesaplanır. Bu işlem her parçacık için yapılır.

6.Adım: Durdurma kontrolü

Durdurma işlemi belirli bir iterasyon yada kabul edilen seviyede uygun bir çözüm ile kontrol edilebilir. Algoritma uygulayıcı tarafından problem için belirlenen durdurma kriteri sağlanmışsa uygulama durdurulur, sağlanmamışsa 2. Adıma tekrar dönülerek yapılan işlemler tekrar edilir.

PSO algoritması yukarıda belirtilen 6 adımda uygulanabilir. PSO'nun süreç adımlarını özet olarak aşağıdaki Algoritma 1 'de sunulmuştur.

Algoritma 1: Parçacık Sürü Optimizasyon Algoritması

1. Parçacıkların başlangıç konumlarını ve hızlarını rastgele belirle.

2. Her bir parçacık için Pbest ve sürü için Gbest hesapla

3. Global en iyi konum (global_best_position) ve global en iyi skoru (global_best_score) başlat.
4. İterasyon sayısı kadar veya durdurma koşulu sağlanana kadar tekrarla:

   a. Her parçacık için:

      i. Parçacığın şu anki konumuna göre fitness hesapla

      ii. Eğer fitness büyük Pbest ise Pbest güncelle

      iii. Eğer fitness büyük Gbest ise Gbest güncelle

   b. Her parçacık için:

      i. Parçacığın hızını güncelle (Eşitlik 1'deki sunulan formüle ile)

      ii. Parçacığın konumunu güncelle: (Eşitlik 2'deki sunulan formüle ile)

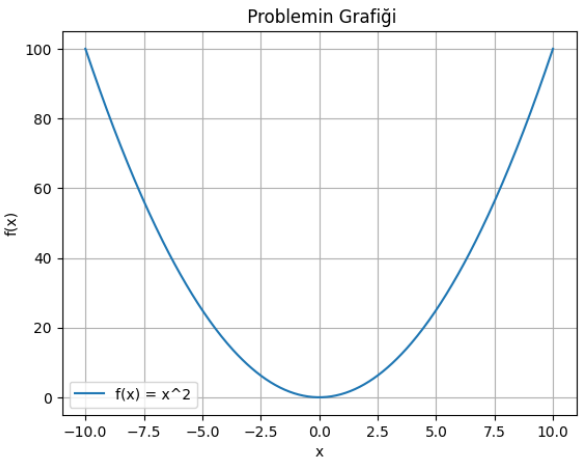
5. Gbest pozisyonu ve Gbest değerini döndür.

PSO uygulama açısından basit parametrelere sahip ve farklı problemlere kolay uygulanabilen bir optimizasyon algoritmasıdır. Çalışmanın devam eden bölümünde X^2 fonksiyonu ve Rastrigin fonksiyonu üzerine PSO uygulanıp sonuçları adım adım sunulmuştur. Ayrıca uygulama için yazılan python kodları verilmiştir.

Örnek 1 : PSO X^2 Örneği

Bu örnekte $f(x) = X^2$ fonksiyonun minimum değerini bulan Bir PSO uygulaması gerçekleştirilmiştir. X^2 tek boyutlu basit bir grafiğe sahip bir işlemdir. Grafik üzerinden her bir parçacığın

hareketini, sürünün dağılımını izlemek için kolay anlaşılır bir örnektir. X^2 grafiği aşağıdaki Şekil 2'de verilmiştir.



Şekil 1: X^2 grafiği

Algoritmanın başlangıç parametreleri;

W = 0,3

$C_1, C_1 \quad = 1{,}5$

Parçacık Sayısı = 5

İterasyon sayısı = 5

X'in çözüm aralığı = $-10 \leq x \leq 10$

Olarak belirlenmiştir.

Python kodu:

```
import numpy as np
import matplotlib.pyplot as plt

# Optimize etmek istediğimiz fonksiyon: f(x) =
x^2
```

```
def cost_function(x):
    return (x ** 2)

# PSO parametreleri
w = 0.3  # Eylemsizlik katsayısı
c1 = 1.5  # Bireysel öğrenme katsayısı
c2 = 1.5  # Sosyal öğrenme katsayısı
n_particles = 5  # Parçacık sayısı
n_iterations = 5  # İterasyon sayısı
dimension = 1  # Tek boyutlu problem (x)
# Parçacıkların başlangıç pozisyonları ve hızları
particles_pos = np.random.uniform(-10, 10,
(n_particles, dimension))  # [-10, 10] aralığında
rastgele başlat
particles_vel = np.random.uniform(-10, 10,
(n_particles, dimension))    # Başlangıç hızları
[-1, 1] aralığında
# Her parçacık için en iyi pozisyon (pbest) ve en
iyi değerler
pbest_pos = np.copy(particles_pos)
pbest_val = np.array([cost_function(p) for p in
pbest_pos])

# Global en iyi pozisyon ve değeri (gbest)
gbest_pos = pbest_pos[np.argmin(pbest_val)]
gbest_val = np.min(pbest_val)

# PSO algoritması
for iteration in range(n_iterations):
    print(f"--- Iterasyon {iteration + 1} ---")
    # Her parçacık için hız ve pozisyon
güncellemeleri
    for i in range(n_particles):
        r1, r2 = np.random.rand(2)  # Rastgele
sayılar
        # Hız güncellemesi
        particles_vel[i] = (w * particles_vel[i]
+
```

```
                                c1 * r1 *
(pbest_pos[i] - particles_pos[i]) +
                                c2 * r2 * (gbest_pos
- particles_pos[i]))

        # Pozisyon güncellemesi
        particles_pos[i] += particles_vel[i]

        # Bu pozisyondaki değeri hesapla
        fitness_value =
cost_function(particles_pos[i])

        # pbest güncellemesi
        if fitness_value < pbest_val[i]:
            pbest_pos[i] = particles_pos[i]
            pbest_val[i] = fitness_value

        # gbest güncellemesi
        if fitness_value < gbest_val:
            gbest_pos = particles_pos[i]
            gbest_val = fitness_value

        print(f"Parçacık {i+1}: Pozisyon =
{particles_pos[i][0]:.4f}, Hız =
{particles_vel[i][0]:.4f}, Maliyet/Fitness =
{fitness_value[0]:.4f}")

    print(f"Pbest Pos:  {[round(val[0], 4) for
val in pbest_pos]}")
    print(f"Pbest Val: {[round(val[0], 4) for val
in pbest_val]}")
    print(f"Gbest: {gbest_pos}\n")
    print(f"Gbest: {gbest_val}\n")

    x = np.linspace(-10, 10, 400)
    y = cost_function(x)

    plt.figure()
    plt.plot(x, y, label="f(x) = x^2")
```

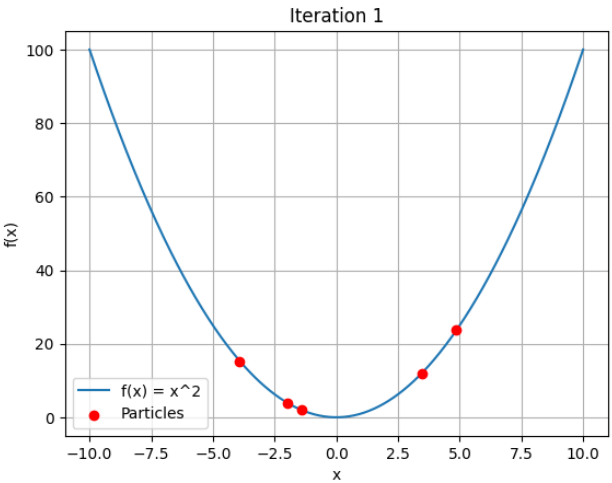
```
    plt.scatter(particles_pos,
cost_function(particles_pos), color='red',
label='Particles', zorder=5)
    plt.title(f"İterasyon {iteration + 1}")
    plt.xlabel("x")
    plt.ylabel("f(x)")
    plt.legend()
    plt.grid(True)
    plt.show()

print(f"Global Best Pozisyon: {gbest_pos}")
print(f"Global Best : {gbest_val}")
```

PSO X^2 örneği Sonuçlarının iteratif olarak incelenmesi:

1. İterasyon: Birinci iterasyon sonucunda elde edilen sonuçlar Tablo 1'te verilmiştir. Tablo 1'deki parçacıkların maliyet fonksiyonu üzerindeki karşılıkları Şekil 3'te grafik üzerinde işaretlenmiştir.

Tablo 1: X^2 örneği 1. iterasyon sonucunda elde edilen değerler

1. İterasyon				
Parçacık	**Pozisyon**	**Hız**	**Fitness**	**Pbest**
1	-3,9121	-12,9176	15,3045	15,3045
2	-1,9918	-0,3136	3,9673	3,9673
3	3,4661	9,0718	12,0137	12,0137
4	4,8675	14,3872	23,6922	23,6922
5	-1,3960	-1,4271	1,9489	1,9489



Şekil 2: X^2 örneği 1. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

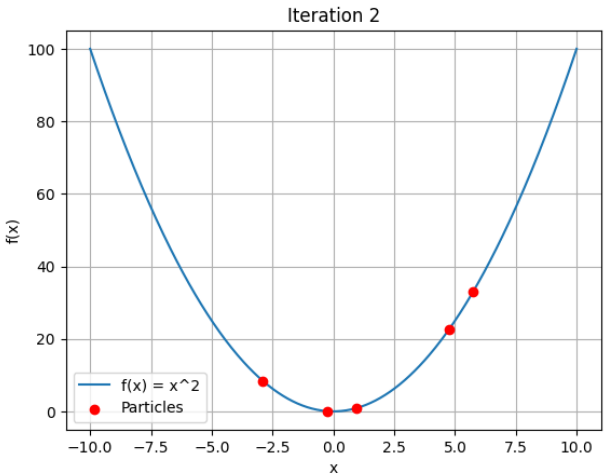
Gbest Pozisyonu: 1,3960

Gbest Fitness: 1,9489

2. İterasyon: İkinci iterasyon sonucunda elde edilen sonuçlar Tablo 2'de verilmiştir. Tablo 2'deki parçacıkların maliyet fonksiyonu üzerindeki karşılıkları Şekil 4'te grafik üzerinde işaretlenmiştir.

Tablo 2: X^2 örneği 2. iterasyon sonucunda elde edilen değerler

2. İterasyon				
Parçacık	**Pozisyon**	**Hız**	**Fitness**	**Pbest**
1	-2,8959	1,0162	8,3864	8,3864
2	-0,2371	1,7547	0,0562	0,0562
3	5,7458	2,2797	33,0138	12,0137
4	4,7452	-0,1223	22,5166	23,6922
5	0,9314	2,3274	0,8675	1,9489



Şekil 3: X^2 örneği 2. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

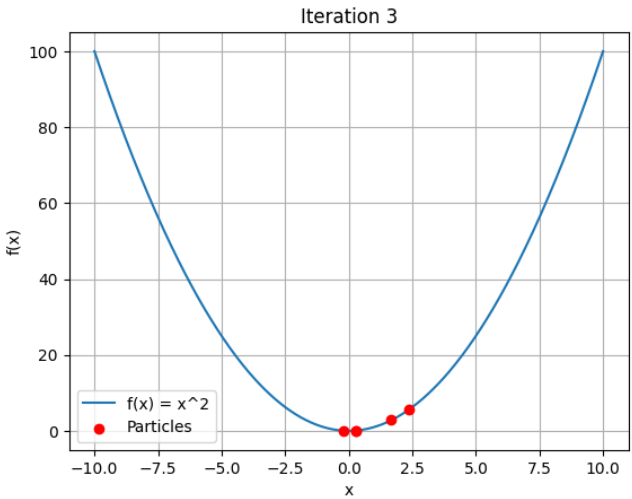
Gbest Pozisyonu : -0,2371

Gbest Fitness : 0,0562

3.İterasyon : Üçüncü iterasyon sonucunda elde edilen sonuçlar Tablo 3'te verilmiştir. Tablo 3'teki parçacıkların maliyet fonksiyonu üzerindeki karşılıkları Şekil 5'te grafik üzerinde işaretlenmiştir.

Tablo 3: X^2 örneği 2. iterasyon sonucunda elde edilen değerler

3. İterasyon				
Parçacık	**Pozisyon**	**Hız**	**Fitness**	**Pbest**
1	1,6759	4,5718	2,8086	2,8087
2	0,2995	0,5365	0,0897	0,0562
3	2,3873	-3,3585	5,6992	5,6993
4	-0,2198	-4,965	0,0483	0,0483
5	0,3004	-0,6311	0,0902	0,0902



Şekil 4: X^2 örneği 3. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

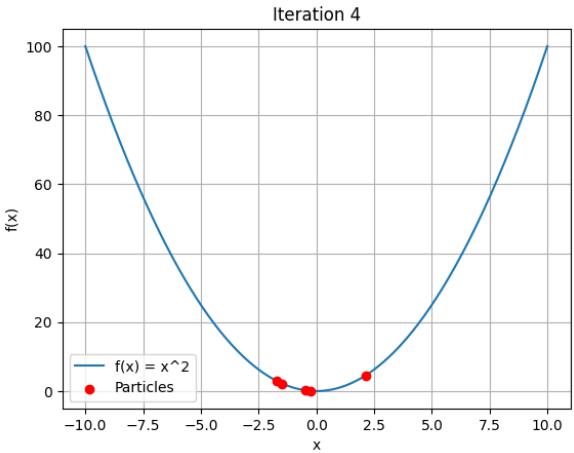
Gbest Pozisyonu : -0,22198

Gbest Fitness : 0,0483

4. İterasyon: Dördüncü iterasyon sonucunda elde edilen sonuçlar Tablo 4'te verilmiştir. Bu iterasyonda Gbest değeri güncellenmemiştir. Tablo 4'teki parçacıkların maliyet fonksiyonu üzerindeki karşılıkları Şekil 6'da grafik üzerinde işaretlenmiştir.

Tablo 4: X^2 örneği 4. iterasyon sonucunda elde edilen değerler

4,.İterasyon				
Parçacık	**Pozisyon**	**Hız**	**Fitness**	**Pbest**
1	2,126	0,4501	4,52	2,8087
2	-0,4655	-0,7649	0,2167	0,0562
3	-1,6946	-4,0819	2,8716	2,8716
4	-1,4809	-1,261	2,1929	0,0483
5	-0,2571	-0,5574	0,0661	0,0661



Şekil 5: X^2 örneği 4. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

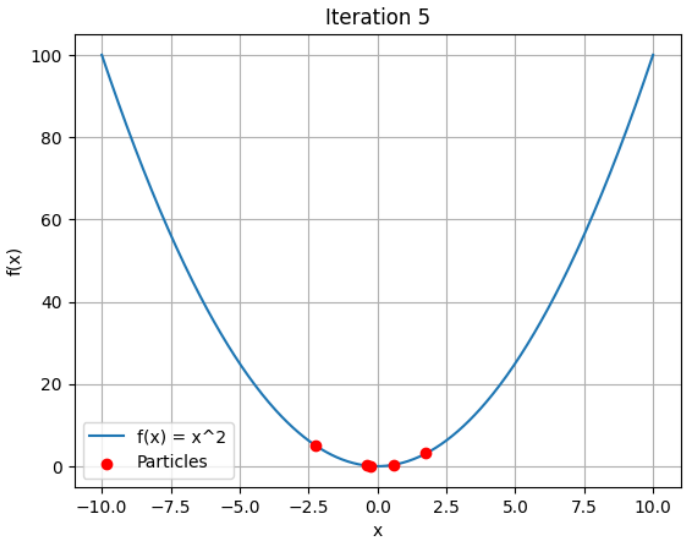
Gbest Pozisyonu : -0,22198

Gbest Fitness : 0,0483

5. İterasyon: Beşinci iterasyon sonucunda elde edilen sonuçlar Tablo 5'te verilmiştir. Bu iterasyonda Gbest değeri güncellenmemiştir. Tablo 5'teki parçacıkların maliyet fonksiyonu üzerindeki karşılıkları Şekil 7'de grafik üzerinde işaretlenmiştir.

Tablo 5: X^2 örneği 5. iterasyon sonucunda elde edilen değerler

5. İterasyon				
Parçacık	**Pozisyon**	**Hız**	**Fitness**	**Pbest**
1	1,7637	-0,3623	3,1107	2,8087
2	-0,393	0,0725	0,1544	0,0562
3	-2,2667	-0,5722	5,1381	2,8716
4	0,6038	2,0847	0,3646	0,0483
5	-0,2552	0,0018	0,0651	0,0651



Şekil 6: X^2 örneği 5. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

Gbest Pozisyonu : -0,2219

Gbest Fitness : 0,0483

Beş iterasyon sonucunda durma kriterine ulaşılmıştır. Beşinci iterasyon sonucunda verilen Gbest pozisyonu -0,2219 algoritmanın erişebildiği en optimum fitness değerine sahip parçacığın pozisyonunu ifade etmektedir. İterasyonlar adım adım takip edilip grafik üzerindeki gösterimleri incelendiğinde sürünün genel olarak nasıl optimize ettiği izlenebilmektedir.

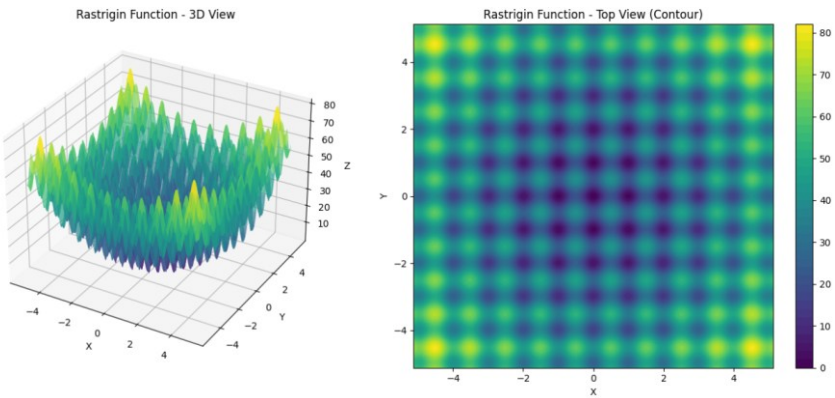
Örnek 2 : PSO Rastrigin fonksiyonu örneği

Rastrigin fonksiyonu özellikle optimizasyon algoritmaları test etmek ve birbirleriyle kıyaslamak için sıklıkla kullanılan bir optimizasyon algoritmasıdır(Omeradzic & Beyer, 2023).

Fonksiyonun matematiksel formülü aşağıdaki Eşitlik 3'te verilmiştir.

$$f(x) = 10 * \mathrm{n} + \sum_{i=1}^{n}(x_i^2 - 10 * \cos(2 * \pi * x_i)) \qquad (3)$$

Rastrigin fonksiyonu yüksek frekanslı sinüs dalgaları sayesinde bir çok yerel minimuma sahip bir fonksiyondur (Korobko vd., 2024). Bu sebeple optimizasyon algoritmaları için zorlayıcı bir fonksiyondur. Global minimum değeri sıfır olan bu fonksiyonun grafiği 3 boyutlu ve tepeden görünüm olarak aşağıdaki Şekil 8'de verilmiştir.



Şekil 7: Rastrigin fonksiyonunun 3 boyutlu grafiği ve tepeden görünümü

Python kodu:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Rastrigin fonksiyonu
def rastrigin(x):
    return 10 * len(x) + sum([xi**2 - 10 * np.cos(2 *
np.pi * xi) for xi in x])
```

```
# PSO parametreleri
w = 0.5  # Eylemsizlik katsayısı
c1 = 1.5  # Bireysel çekim
c2 = 1.5  # Sürü çekimi
n_particles = 5  # Parçacık sayısı
n_iterations = 10  # İterasyon sayısı
dimension = 2  # Problem boyutu (2D uzay)

# Parçacıkların başlangıç pozisyonları ve hızları
particles_pos = np.random.uniform(-5.12, 5.12,
(n_particles, dimension))
particles_vel = np.random.uniform(-1, 1, (n_particles,
dimension))

# Her bir parçacık için en iyi pozisyonlar (pbest) ve
global en iyi pozisyon (gbest)
pbest_pos = np.copy(particles_pos)
pbest_val = np.array([rastrigin(p) for p in
pbest_pos])
gbest_pos = pbest_pos[np.argmin(pbest_val)]  # En iyi
global pozisyon
gbest_val = np.min(pbest_val)  # Global en iyi fitness
değeri
pabestKayit=0
gbestKayit=0
# Rastrigin fonksiyonu yüzeyi için grid oluştur
x = np.linspace(-5.12, 5.12, 400)
y = np.linspace(-5.12, 5.12, 400)
X, Y = np.meshgrid(x, y)
Z = 10 * 2 + (X**2 - 10 * np.cos(2 * np.pi * X)) +
(Y**2 - 10 * np.cos(2 * np.pi * Y))

# PSO'nun grafiksel gösterimi
for iteration in range(n_iterations):
    # Her iterasyon için iki grafik oluştur (3D yüzey
ve contour)
    # 1. Grafik: 3D Yüzey
    fig = plt.figure(figsize=(12, 8))  # Yüksekliği
biraz arttırarak alttaki metne yer açıyoruz

    # Sol tarafta 3D grafik
    ax1 = fig.add_subplot(121, projection='3d')
```

```
    ax1.plot_surface(X, Y, Z, cmap='viridis',
alpha=0.6)

    # Parçacıkların pozisyonları ve hızlarının
güncellenmesi
    for i in range(n_particles):
        # Hızın güncellenmesi
        r1, r2 = np.random.rand(2)
        particles_vel[i] = (w * particles_vel[i] +
                            c1 * r1 * (pbest_pos[i] -
particles_pos[i]) +
                            c2 * r2 * (gbest_pos -
particles_pos[i]))

        # Konumun güncellenmesi
        particles_pos[i] += particles_vel[i]

        # Yeni pozisyonun fitness değerini hesapla
        fitness_value = rastrigin(particles_pos[i])

        # pbest güncellemesi
        if fitness_value < pbest_val[i]:
            pbest_pos[i] = particles_pos[i]
            pbest_val[i] = fitness_value
        # gbest güncellemesi
        if fitness_value < gbest_val:
            gbest_pos = particles_pos[i]
            gbest_val = fitness_value

        # Parçacık pozisyonlarını 3D yüzey üzerinde
çizdir
        ax1.scatter(particles_pos[i][0],
particles_pos[i][1], rastrigin(particles_pos[i]),
                    color='r', s=50, label=f'P{i+1}' if
iteration == 0 and i == 0 else "")

        try:
          pabestKayit=pbest_val[i]
          gbestKayit=gbest_val
        except:
          pabestKayit=0
          gbestKayit=0
```

```
    # Global en iyi pozisyonu (gbest) göster
    ax1.scatter(gbest_pos[0], gbest_pos[1],
rastrigin(gbest_pos),
               color='b', s=100, marker='*',
label="Global Best")

    # Eksen ayarları
    ax1.set_title(f"3D Surface Plot - İterasyon
{iteration + 1}")
    ax1.set_xlabel('X ')
    ax1.set_ylabel('Y ')

    ax1.legend(loc='upper right')

    # 2. Grafik: Contour Plot (Tepeden Görünüm)
    ax2 = fig.add_subplot(122)
    ax2.contourf(X, Y, Z, levels=50, cmap='viridis')

    # Parçacıkların pozisyonlarını contour grafiğinde
çizdir
    for i in range(n_particles):
        ax2.scatter(particles_pos[i][0],
particles_pos[i][1], color='r', s=50, label=f'P{i+1}'
if iteration == 0 and i == 0 else "")

    # Global en iyi pozisyonu (gbest) göster
    ax2.scatter(gbest_pos[0], gbest_pos[1], color='b',
s=100, marker='*', label="Global Best")

    # Contour grafiği için ayarlar
    ax2.set_title(f"Contour Plot - İterasyon
{iteration + 1}")

    ax2.legend(loc='upper right')

    # En iyi pbest ve gbest değerlerini göster
    plt.figtext(0.5, 0.01, f"İterasyon {iteration + 1}
- G Best (gbest): {gbest_val:.4f} - "
                           f"P Bests (pbest):
{[round(val, 4) for val in pbest_val]}",
                ha="center", fontsize=12,
bbox={"facecolor":"orange", "alpha":0.5, "pad":5})
```

```
# Grafikleri göster
plt.tight_layout()
plt.show()
```

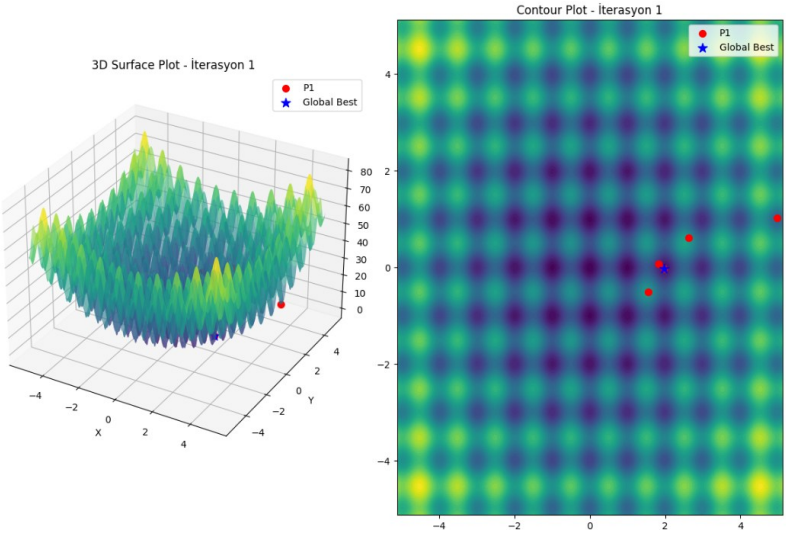
PSO Rastrigin fonksiyonu örneği sonuçlarının iteratif olarak incelenmesi:

Rastrigin fonksiyonu üzerinde 5 parçacıkla 10 iterasyon sonucunda elde edilen sonuçlar her bir iterasyondaki değerleri ve iterasyon sonundaki parçacıkların fonksiyon grafiği üzerindeki pozisyonları sırasıyla sunulmuştur.

1. İterasyon: Birinci iterasyon sonucundaki değerler Tablo 6'da ve parçacıkların iterasyon sonundaki pozisyonları Şekil 9'da verilmiştir.

Tablo 6: Rastrigin fonksiyonu örneği 1. iterasyon sonucunda elde edilen değerler

1. İterasyon						
Parçacık	**Pozisyon_x**	**Pozisyon_y**	**Hız_x**	**Hız_y**	**Fitness**	**Pbest**
1	1,8160	0,0744	-0,1629	0,1045	10,3481	4,1840
2	1,9757	-0,0204	-0,7811	2,5928	4,1019	4,1019
3	4,9656	1,0198	9,3406	3,5961	26,0074	26,0074
4	1,5563	-0,5054	-0,1003	-1,2228	42,0522	30,8344
5	2,6227	0,6109	-0,5446	-0,2679	42,0944	18,6054



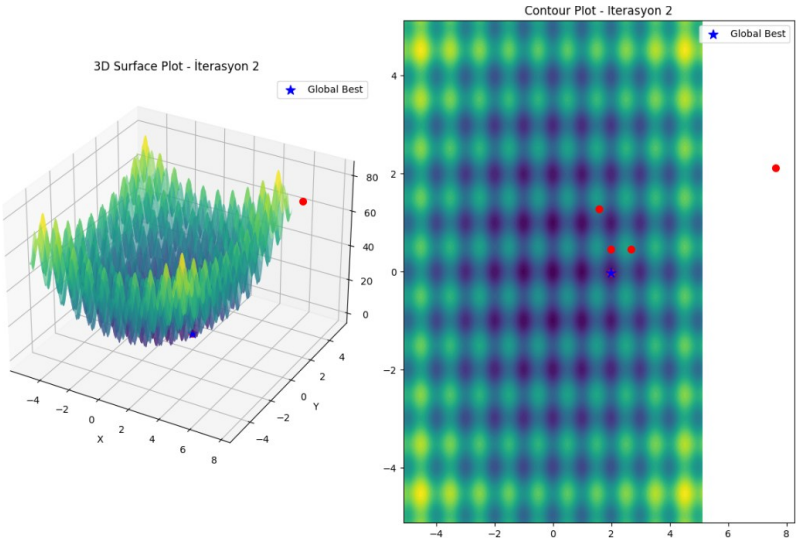
Şekil 8: Rastrigin örneği 5. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

Gbest : 4,1840

<u>2. İterasyon:</u> Birinci iterasyon sonucundaki değerler Tablo 7'de ve parçacıkların iterasyon sonundaki pozisyonları Şekil 10'da verilmiştir.

Tablo 7: Rastrigin fonksiyonu örneği 2. iterasyon sonucunda elde edilen değerler

2. İterasyon						
Parçacık	**Pozisyon_x**	**Pozisyon_y**	**Hız_x**	**Hız_y**	**Fitness**	**Pbest**
1	1,9826	-0,0210	0,1666	-0,0953	4,0777	4,0777
2	1,5869	1,2759	-0,3889	1,2962	34,3111	4,1019
3	7,6351	2,1198	2,6695	1,0999	82,0960	26,0074
4	1,9913	0,4585	0,4350	0,9639	23,8516	23,8516
5	2,6826	0,4650	0,0599	-0,1459	41,2821	18,6054



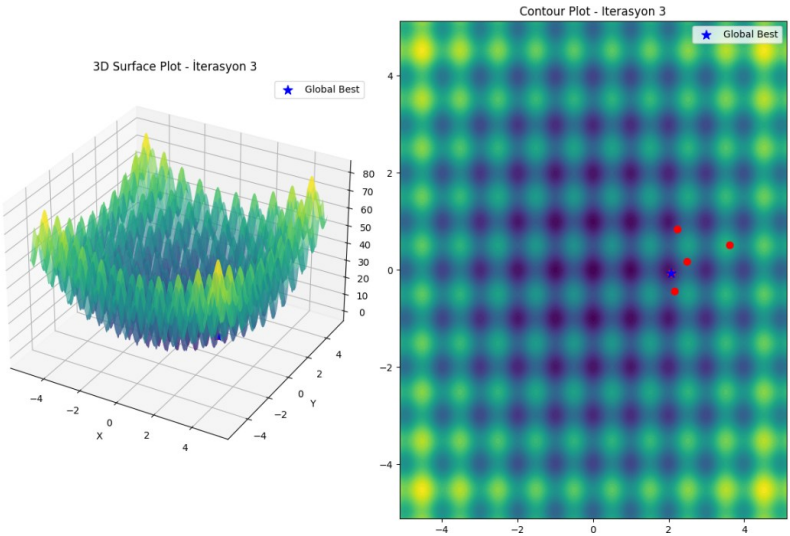
Şekil 9: Rastrigin örneği 2. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

Gbest : 4,0777

<u>3. İterasyon:</u> Birinci iterasyon sonucundaki değerler Tablo 8'de ve parçacıkların iterasyon sonundaki pozisyonları Şekil 11'de verilmiştir.

Tablo 8: Rastrigin fonksiyonu örneği 3. iterasyon sonucunda elde edilen değerler

3. İterasyon						
Parçacık	**Pozisyon_x**	**Pozisyon_y**	**Hız_x**	**Hız_y**	**Fitness**	**Pbest**
1	2,0659	-0,0687	0,0833	-0,0477	6,0343	4,0777
2	2,1578	-0,4305	0,5710	-1,7064	28,4312	4,1019
3	3,6050	0,5137	-4,0302	-1,6060	51,1256	26,0074
4	2,2231	0,8389	0,2319	0,3804	18,6680	18,6680
5	2,4718	0,1765	-0,2108	-0,2885	31,5274	18,6054



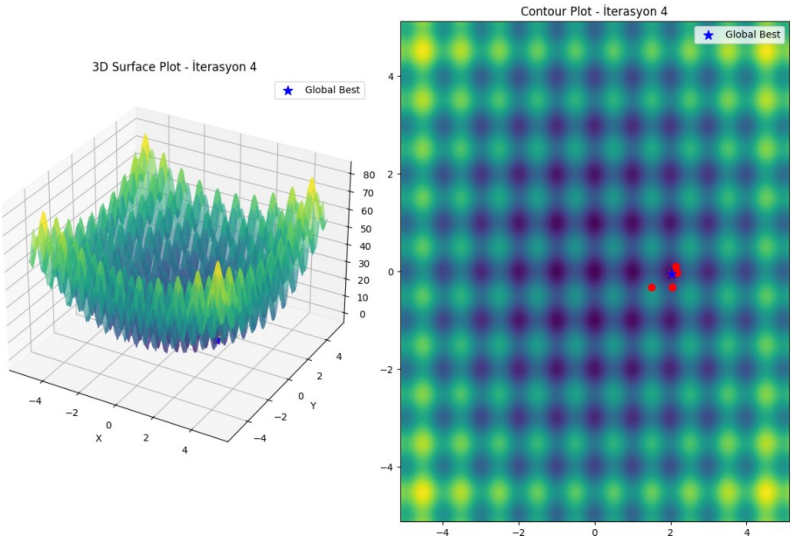
Şekil 10: Rastrigin örneği 3. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

Gbest : 4,0777

<u>4. İterasyon:</u> Birinci iterasyon sonucundaki değerler Tablo 9'da ve parçacıkların iterasyon sonundaki pozisyonları Şekil 12'de verilmiştir.

Tablo 9: Rastrigin fonksiyonu örneği 4. iterasyon sonucunda elde edilen değerler

4. İterasyon						
Parçacık	**Pozisyon_x**	**Pozisyon_y**	**Hız_x**	**Hız_y**	**Fitness**	**Pbest**
1	2,0260	-0,0458	-0,0400	0,0229	4,6506	4,0777
2	2,0522	-0,3226	-0,1056	0,1080	19,2517	4,1019
3	1,4963	-0,3174	-2,1086	-0,8312	36,4476	26,0074
4	2,1335	0,1066	-0,0897	-0,7322	10,0405	10,0405
5	2,1646	-0,0216	-0,3072	-0,1981	9,6665	9,6665



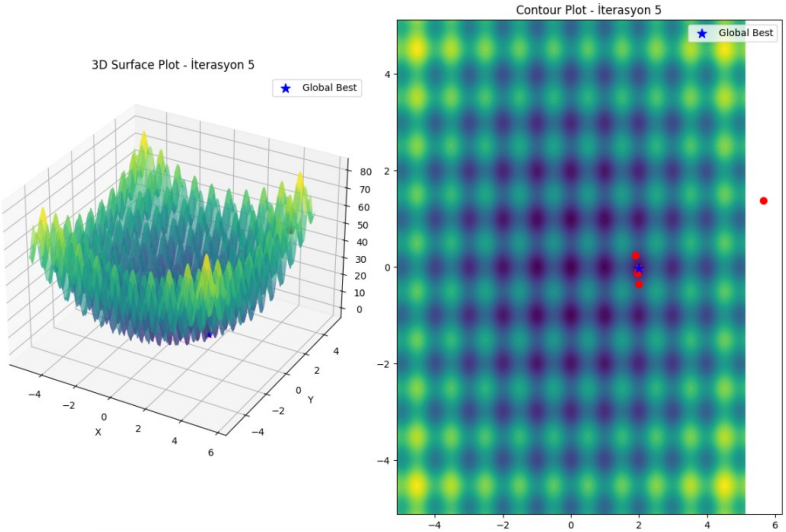
Şekil 11: Rastrigin örneği 4. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

Gbest : 4,0777

5. İterasyon: Birinci iterasyon sonucundaki değerler Tablo 10'da ve parçacıkların iterasyon sonundaki pozisyonları Şekil 13'te verilmiştir.

Tablo 10: Rastrigin fonksiyonu örneği 5. iterasyon sonucunda elde edilen değerler

5. İterasyon						
Parçacık	**Pozisyon_x**	**Pozisyon_y**	**Hız_x**	**Hız_y**	**Fitness**	**Pbest**
1	1,9899	-0,0251	-0,0361	0,0207	4,1050	4,0777
2	1,8881	0,2429	-0,1641	0,5655	15,5493	4,1019
3	5,6605	1,3817	4,1642	1,6991	66,6424	26,0074
4	1,9879	-0,3519	-0,1456	-0,4586	20,0795	10,0405
5	1,9618	-0,1216	-0,2028	-0,1000	6,9297	6,9297



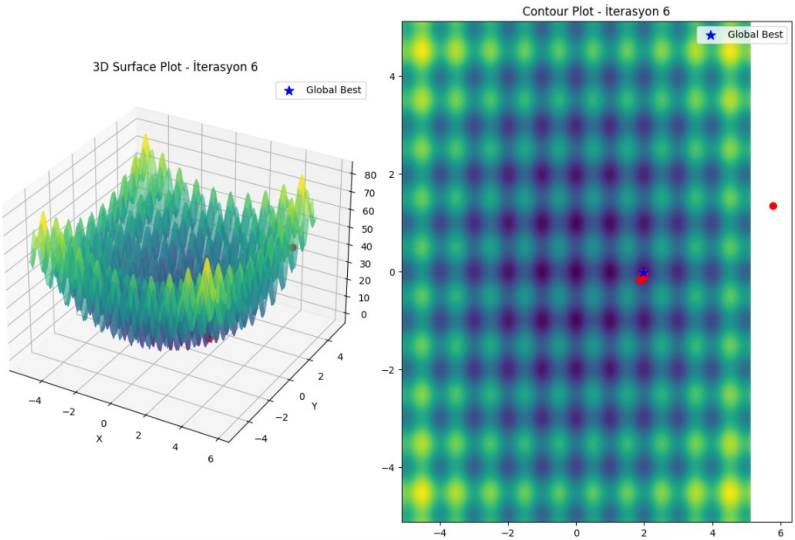
Şekil 12: Rastrigin örneği 5. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

Gbest : 4,0777

<u>6. İterasyon:</u> Birinci iterasyon sonucundaki değerler Tablo 11'de ve parçacıkların iterasyon sonundaki pozisyonları Şekil 14'te verilmiştir.

Tablo 11: Rastrigin fonksiyonu örneği 6. iterasyon sonucunda elde edilen değerler

6. İterasyon						
Parçacık	**Pozisyon_x**	**Pozisyon_y**	**Hız_x**	**Hız_y**	**Fitness**	**Pbest**
1	1,9630	-0,0098	-0,0269	0,0154	4,1416	4,0777
2	1,9851	-0,0455	0,0970	-0,2884	4,3921	4,1019
3	5,7799	1,3559	0,1194	-0,0258	59,5508	26,0074
4	1,9940	-0,1155	0,0061	0,2364	6,5182	6,5182
5	1,8605	-0,1647	-0,1013	-0,0431	11,9840	6,9297



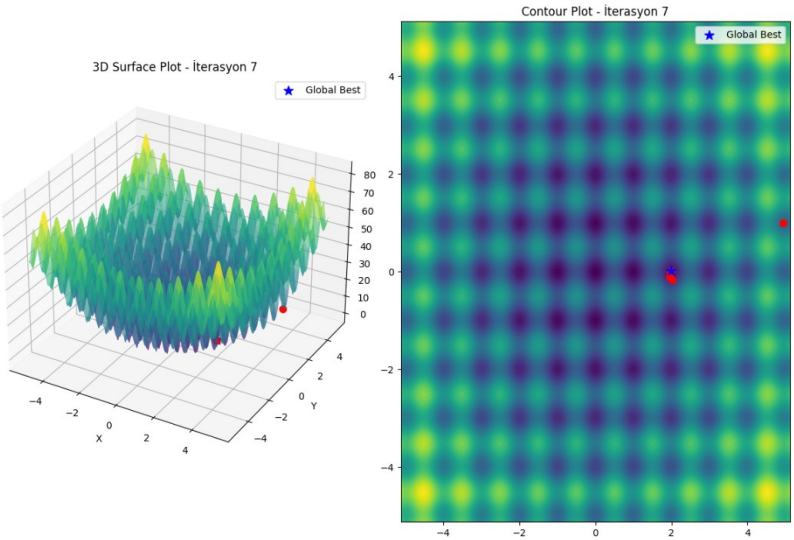
Şekil 13: Rastrigin örneği 6. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

Gbest : 4,0777

7. İterasyon: Birinci iterasyon sonucundaki değerler Tablo 12'de ve parçacıkların iterasyon sonundaki pozisyonları Şekil 15'te verilmiştir.

Tablo 12: Rastrigin fonksiyonu örneği 7. iterasyon sonucunda elde edilen değerler

7. İterasyon						
Parçacık	**Pozisyon_x**	**Pozisyon_y**	**Hız_x**	**Hız_y**	**Fitness**	**Pbest**
1	1,9768	-0,0177	0,0138	-0,0079	4,0757	4,0757
2	2,0233	-0,1617	0,0383	-0,1162	8,9588	4,1019
3	4,9415	0,9959	-0,8384	-0,3599	26,0815	26,0074
4	1,9947	0,0160	0,0007	0,1315	4,0350	4,0350
5	1,9433	-0,1016	0,0828	0,0631	6,3849	6,3849



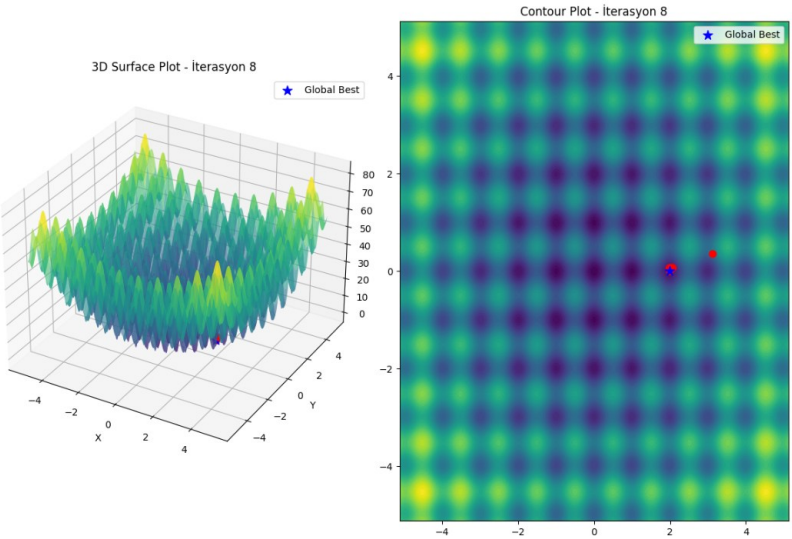
Şekil 14: Rastrigin örneği 7. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

Gbest : 4,0757

8. İterasyon: Birinci iterasyon sonucundaki değerler Tablo 13'te ve parçacıkların iterasyon sonundaki pozisyonları Şekil 16'da verilmiştir.

Tablo 13: Rastrigin fonksiyonu örneği 8. iterasyon sonucunda elde edilen değerler

8. İterasyon						
Parçacık	**Pozisyon_x**	**Pozisyon_y**	**Hız_x**	**Hız_y**	**Fitness**	**Pbest**
1	1,9943	-0,0017	0,0175	0,0159	3,9842	3,9842
2	1,9688	0,0572	-0,0545	0,2189	4,7094	4,1019
3	3,1264	0,3607	-1,8151	-0,6353	29,3037	26,0074
4	1,9950	0,0774	0,0003	0,0614	5,1500	4,0350
5	2,0565	0,0706	0,1132	0,1723	5,8267	5,8267



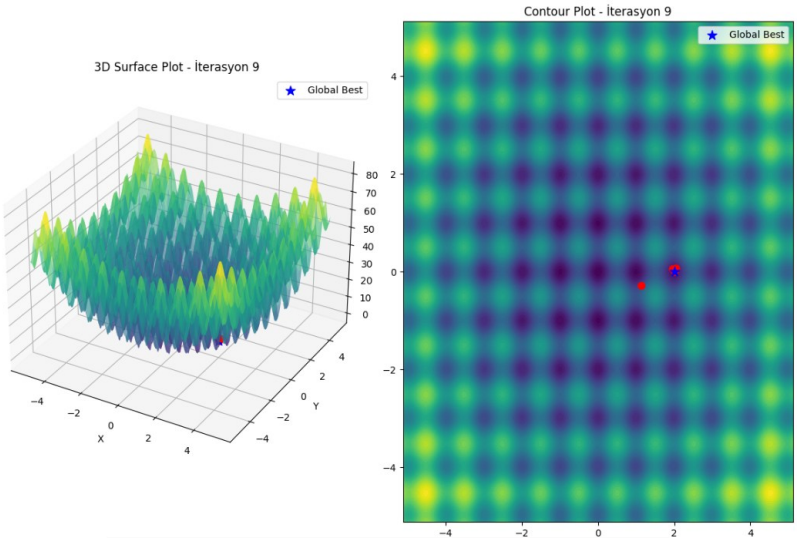
Şekil 15: Rastrigin örneği 8. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

Gbest : 3,9842

<u>9. İterasyon:</u> Birinci iterasyon sonucundaki değerler Tablo 14'te ve parçacıkların iterasyon sonundaki pozisyonları Şekil 17'de verilmiştir.

Tablo 14: Rastrigin fonksiyonu örneği 9. iterasyon sonucunda elde edilen değerler

9. İterasyon						
Parçacık	**Pozisyon_x**	**Pozisyon_y**	**Hız_x**	**Hız_y**	**Fitness**	**Pbest**
1	2,0031	0,0063	0,0088	0,0080	4,0219	3,9842
2	1,9587	0,0582	-0,0101	0,0010	4,8361	4,1019
3	1,1410	-0,2816	-1,9854	-0,6422	17,0291	17,0291
4	1,9999	0,0146	0,0050	-0,0628	4,0422	4,0350
5	2,0405	0,0693	-0,0160	-0,0013	5,4247	5,4247



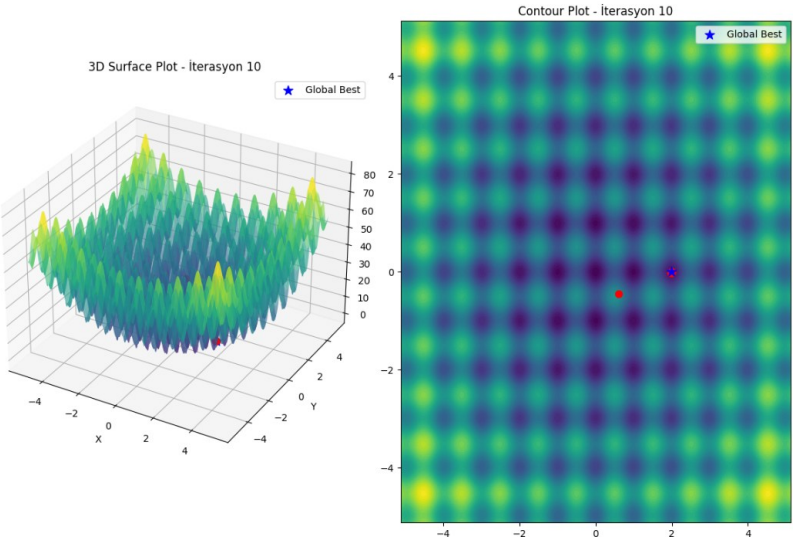
Şekil 16: Rastrigin örneği 9. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

Gbest : 3,9842

<u>10. İterasyon:</u> Birinci iterasyon sonucundaki değerler Tablo 15'te ve parçacıkların iterasyon sonundaki pozisyonları Şekil 18'de verilmiştir.

Tablo 15: Rastrigin fonksiyonu örneği 10. iterasyon sonucunda elde edilen değerler

10. İterasyon						
Parçacık	**Pozisyon_x**	**Pozisyon_y**	**Hız_x**	**Hız_y**	**Fitness**	**Pbest**
1	2,0051	0,0081	0,0021	0,0019	4,0388	3,9842
2	1,9941	-0,0021	0,0354	-0,0603	3,9841	3,9841
3	0,6096	-0,4516	-0,5314	-0,1700	37,8358	17,0291
4	1,9959	-0,0245	-0,0040	-0,0391	4,1058	4,0350
5	1,9680	-0,0306	-0,0725	-0,0999	4,2593	4,2593



Şekil 17: Rastrigin örneği 10. iterasyon sonucunda parçacıkların grafik üzerinde gösterimi

Gbest : 3,9841

PSO algoritması ile Rastrigin fonksiyonunun optimize edilme örneği 10. İterasyon sonunda durma kriterine gelip sonlandırılmıştır. 2. Parçacık 10. iterasyonda 3,9841 fitness değerine gelerek sürünün en uygun pozisyonuna ulaşmıştır.

PSO algoritması ilk olarak X^2 fonksiyonu üzerinde beş parçacıkla 5 iterasyon boyunca optimize edilerek her bir iterasyon sonucundaki parçacıkların pozisyonları, hızları, fitness'ları ve Pbest değerleri tablolar halinde verilmiş ayrıca anlık pozisyonlar grafik üzerinde işaretlenerek sürünün hareketi gösterilmiştir. Aynı şekilde ikinci örnekte de Rastrigin fonksiyonu üzerinde her bir iterasyon sonucu tablo ve grafikler üzerinde sunularak sürünün genel hareketi izlenmesi sağlanmıştır.

KAYNAKÇA

Beheshti, Z., & Shamsuddin, S. M. H. (2013). A review of population-based meta-heuristic algorithms. *Int. j. adv. soft comput. appl*, *5*(1), 1-35.

Bülbül, M. A. (2023). Kuru Fasulye Tohumlarının Çok Sınıflı Sınıflandırılması İçin Hibrit Bir Yaklaşım. *Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *13*(1), 33-43. https://doi.org/10.21597/JIST.1185949

Çetintaş, D., Tuncer, T., Makalesi, A., Research, /, & Bilgileri, A. M. (2023). Göz Özelliklerinin LSTM-PSO Modeli kullanılarak Otizm Sınıflandırılması. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, *14*(4), 563-570. https://doi.org/10.24012/DUMF.1327654

Derdiman, M. K. (2022). AYRIK PSO ALGORİTMASI İLE KONSOL DÖŞEMELERİN GÜVENİLİRLİK TABANLI KESİT OPTİMİZASYONU. *Mühendislik Bilimleri ve Tasarım Dergisi*, *10*(3), 987-999. https://doi.org/10.21923/JESD.952838

Kaveh, A., & Nasrollahi, A. (2013). ENGINEERING DESIGN OPTIMIZATION USING A HYBRID PSO AND HS ALGORITHM. *ASIAN JOURNAL OF CIVIL ENGINEERING (BHRC)*, *14*(2), 201-223. www.SID.ir

Kennedy, J., & Eberhart, R. (t.y.). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, *4*, 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-international conference on neural networks*, *4*, 1942-1948.

Kokash, N. (2005). An introduction to heuristic algorithms. *Department of Informatics and Telecommunications*, 1-8.

Korobko, A. S., Gazizov, T. T., & Moskalev, M. (2024). Comparison of Evolutionary Optimization Methods on the Example of Rastrigin Function. *Proceedings - 2024 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology, USBEREIT 2024*, 49-53. https://doi.org/10.1109/USBEREIT61901.2024.10584012

Lale Zeynelgil, H., Üniversitesi, İ. T., Bölümü, M., & Geçmişi, M. (2023). Dağıtım Sistemlerinin Yeniden Yapılandırılması Problemine PSO Algoritmasının Uygulanması. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, *14*(2), 215-227. https://doi.org/10.24012/DUMF.1231678

Marini, F., & Walczak, B. (2015a). Particle swarm optimization (PSO). A tutorial. *Chemometrics and Intelligent Laboratory Systems*, *149*, 153-165. https://doi.org/10.1016/J.CHEMOLAB.2015.08.020

Marini, F., & Walczak, B. (2015b). Particle swarm optimization (PSO). A tutorial. *Chemometrics and Intelligent Laboratory Systems*, *149*, 153-165. https://doi.org/10.1016/J.CHEMOLAB.2015.08.020

Omeradzic, A., & Beyer, H. G. (2023). Progress analysis of a multi-recombinative evolution strategy on the highly multimodal

Rastrigin function. *Theoretical Computer Science*, *978*, 114179. https://doi.org/10.1016/J.TCS.2023.114179

Özmen, M., Delice, Y., Kizilkaya Aydoğan, E., Mühendisliği, E., Üniversitesi, E., & ve Organizasyon Bölümü, Y. (2018). Telekomünikasyon Sektöründe PSO ile Müşteri Bölümlemesi. *Bilişim Teknolojileri Dergisi*, *11*(2), 163-173. https://doi.org/10.17671/GAZIBTD.368460

Ulusoy, A., & Güneş, M. (2019). MOBİL ROBOT KOLUNUN PSO İLE STABİLİZASYONU. *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, *22*(4), 288-297. https://doi.org/10.17780/KSUJES.600310

Yi, L. (2016). Study on an Improved PSO Algorithm and its Application for Solving Function Problem. *International Journal of Smart Home*, *10*(3), 51-62. https://doi.org/10.14257/ijsh.2016.10.3.06

Yılmaz, Z. N., Metin, N. A., & Lüy, M. (2022). Su Altı Araçlarında Servo Motor Kontrolünün PSO ile Optimizasyonu. *International Journal of Engineering Research and Development*, *14*(2), 743-751. https://doi.org/10.29137/UMAGD.1124999

BÖLÜM VI

Patates Yaprak Görüntüleri İle Hastalık Teşhisi: Vision Transformer Ve VGG16 Derin Öğrenme Yöntemlerinin Karşılaştırması

Alper Talha KARADENİZ[1]

1. Giriş

1.1. Araştırmanın Amacı

Dünya nüfusunun hızla artmasıyla birlikte, tarım sektörü, daha verimli, sürdürülebilir ve yenilikçi yaklaşımlara ihtiyaç duymaktadır (Karadeniz, Çelik, et al., 2023). Patates gibi temel tarım ürünlerinde karşılaşılan hastalıklar, ürün kayıplarına ve ekonomik zararlara neden olmaktadır. Bu hastalıkların erken teşhisi, hem üretim kalitesini artırmak hemde tarımsal sürdürülebilirliği desteklemek için hayati bir öneme sahiptir. Hastalık tespitinde kullanılan geleneksel yöntemler zaman alıcı, maliyetli ve hata oranı yüksek olabilmektedir (Fiers et al., 2012). Bu yüzden hastalık tespitini otomatikleştirmek önemlidir.

[1] Dr. Öğr. Üyesi Samsun Üniversitesi Mühendsilik ve Doğa Bilimleri Fakültesi Yazılım Mühendisliği Bölümü

Bu çalışmanın amacı, derin öğrenme tabanlı iki farklı model olan Vision Transformer (ViT) ve VGG16'nın, patates yaprak görüntülerinden hastalıklarının sınıflandırılması üzerindeki performanslarını karşılaştırmaktır. Çalışma, bu iki modelin sınıflandırma doğruluğu, genelleme yeteneği ve eğitim süreleri gibi metrikler açısından değerlendirilmesini hedeflemektedir.

1.2. Motivasyon

Derin öğrenme, gelişen teknoloji ile birlikte, sağlık, ulaşım, finans ve tarım gibi birçok alanda yeniliklere imza atmıştır. Özellikle tarımda kullanılan yapay zeka tabanlı çözümler, üretim verimliliği, maliyet azaltma, bitki hastalıklarının tespiti başta olmak üzere büyük potansiyel sunmaktadır (Li et al., 2021). Patates yaprak hastalıklarının teşhisi için geliştirilen modeller, yalnızca üretim kayıplarını azaltmakla kalmaz, aynı zamanda çiftçilerin daha bilinçli kararlar almasına yardımcı olmaktadır.

Çalışmada şu temel sorulara yanıt aramaktadır:

- Geleneksel CNN tabanlı modeller (VGG16) ile transformer tabanlı modeller (ViT) arasındaki performans farkı nedir?
- Hangi model küçük ölçekli tarımsal veri setlerinde daha iyi performans gösterir?
- Her iki modelin doğruluk, genelleme yeteneği açısından avantajları ve dezavantajları nelerdir?

Bu sorulara yanıt bulmak, tarımsal görüntü sınıflandırma uygulamaları için daha etkili yöntemler geliştirilmesine katkıda bulunacaktır.

1.3. Problem Tanımı

Patates yapraklarındaki hastalıkları teşhis etmek, genellikle uzman görüşüne dayalı görsel inceleme veya laboratuvar deneyleri ile yapılmaktadır. Ancak görsel incelemede bazı hastalık belirtileri diğerleriyle karışabilir veya laboratuvar deneyleri hem maliyetli hemde zaman alıcıdır. Ayrıca büyük ölçekli tarım arazilerinde her

bitkiyi tek tek incelemek mümkün olmamaktadır. Bunların dışında, kırsal bölgelerde hastalık teşhisinde deneyimli uzmanlara erişim sınırlıdır.

Çalışmada, bu sorunları aşabilmek adına patates yaprak görüntülerindeki hastalıkların otomatik olarak sınıflandırılması hedeflenmektedir.

1.4. Literatür Taraması

Tarımda, derin öğrenme tabanlı çözümler üzerine yapılan çalışmalar, bu alanın hızla büyüdüğünü göstermektedir. Bu çalışmada kullanılan Vision Transformer (ViT) ve VGG16 modelleri üzerine odaklanarak önceki literatürden bazı önemli bulgular sunulmuştur.

CNN tabanlı bir modelle yapılan çalışmada 13 farklı bitki hastalığını sınıflandırmıştır. Model, %96 doğruluk elde ederek derin öğrenmenin tarımda uygulanabilirliğini göstermektedir. Bu çalışma, daha sonra geliştirilecek olan VGG16 gibi modellerin temelini oluşturmuştur (Sladojevic et al., 2016). Bir başka çalışmada derin öğrenme kullanılarak, tarım için uygun maliyetli ve doğru bitki hastalığı tespiti için yoğun bağlantılı ağlar ve IoT cihazlarında çalışan hafif modeller önerilmektedir (Ale et al., 2019). Dosovitskiy yaptığı çalışmada ViT'nin büyük veri setlerinde etkili bir sınıflandırma yöntemi olduğunu göstermektedir. Transformer tabanlı yaklaşımlar, küçük veri setleriyle çalışırken transfer öğrenim ve veri artırma tekniklerinden faydalanmalıdır (Dosovitskiy, 2020). Transformer tabanlı modeller ile yapılan bir diğer çalışma, tarımsal görüntü sınıflandırmada kullanılabilirliğini test etmektedir. Çalışma, ViT'nin dikkat mekanizmalarının karmaşık görüntü desenlerini anlamada üstün olduğunu ortaya koymaktadır (Wang et al., 2021). Literatürde Hibrit CNN modelleri de kullanılmaktadır. Bu çalışma, hibrit CNN modelleri ve transfer öğrenim teknikleri ile %94.70 doğruluk oranına ulaşarak, hastalık yönetimi ve gıda güvenliğine katkı sağlamaktadır (Sharma ve Sharma, 2024).

Literatürde patates hastalıklarının yaprak görüntülerinden otomatik tespiti için yapılan derin öğrenme çalışmaları bulunmaktadır.

Ferentinos yaptığı çalışmada, transfer öğrenmenin etkinliğini vurgulamış ve çeşitli CNN mimarilerini kıyaslamıştır. Patates yaprak hastalıklarını teşhis eden bir modelle %99 doğruluk elde etmiştir (Ferentinos, 2018). Transfer öğrenimini kullanan bir başka çalışmada patates yaprak hastalıklarının tespitinde %95 doğruluk oranı elde etmiştir. Araştırmada, VGG-16 tabanlı transfer öğrenim modeli kullanılarak küçük veri setleriyle başarılı sonuçlar alınmıştır (Rozaqi ve ark., 2021).

Too ve ark., farklı CNN modellerinin tarımsal veri setlerinde karşılaştırarak, VGG16'nın küçük veri setlerinde daha tutarlı bir performans sergilediğini belirtmiştir. Patates yapraklarındaki erken yanıklık ve bakteriyel leke gibi hastalıkları sınıflandırmak için CNN tabanlı bir model geliştirilmiştir. Bu çalışma, %94 doğruluk oranıyla tarımsal veri setlerinde derin öğrenmenin başarısını göstermektedir (Too ve ark., 2019).

Patates yaprak hastalıklarının erken teşhisi için önerilen yöntem, önceden eğitilmiş CNN modelleriyle %97,4 doğruluk oranına ulaşmıştır. Bu, hastalık yönetiminde etkili bir araç olarak öne çıkmaktadır (Pasalkar et al., 2023). CNN modeli ile hastalık tespiti üzerine yapılan bir başka çalışmada, erken yanıklık ve geç yanıklık gibi hastalıklar %97 doğruluk oranı ile sınıflandırılmıştır (Haidari & Kumar, 2023).

Asif ve ark, %97 doğruluk oranı ile hastalıkların görüntü işleme yöntemleri kullanılarak tespit edilmesini önermekmektedir (Asif ve ark., 2020). Yapılan bir başka görüntü işleme çalışmasında ise %95 doğruluk oranı ile erken yanıklık ve geç yanıklık gibi hastalıkların tespiti yapılmıştır (Husna, 2024).

Singh ve ark., CNN ve rastgele orman modellerini birleştirerek, patates yaprak hastalıklarının tespiti için %89.26

doğruluk oranı ile etkili bir yöntem geliştirmiştir (Singh ve ark., 2022).

Vision Transformer (ViT) modelleri ile yapılan çalışma, patates yaprak hastalıklarının %99.55 doğruluk oranı ile sınıflandırılmasını sağlamıştır. Bu, tarımsal teknolojide yeni bir standart sunmaktadır (Adhikari, 2024).

Sholihati ve ark., VGG16 ve VGG19 modellerini kullanarak %97 doğruluk oranı ile patates yaprak hastalıklarını sınıflandırmıştır. Bu, tarımsal üretimde sürdürülebilirliği artırmak için etkili bir araç olarak önerilmektedir (Sholihati ve ark., 2020).

Nawaz ve ark., EfficientNet-V2 tabanlı EfficientPNet modeli, patates yaprak hastalıklarını %98.12 doğruluk oranı ile sınıflandırarak, tarımsal verimliliği artırmayı hedeflemiştir (Nawaz ve ark., 2023).

1.5 Çalışmanın Katkısı

Bu çalışma, Vision Transformer (ViT) ve VGG16 modellerinin patates yaprak hastalıklarını sınıflandırmadaki performansını karşılaştırarak literatüre şu katkıları sağlamayı amaçlamaktadır:

- Tarımsal veri setlerinde transformer tabanlı modellerin (ViT) ve CNN tabanlı modellerin (VGG16) etkinliğini doğrudan kıyaslamak.
- Küçük veri setlerinde transfer öğrenme ve veri artırma yöntemlerinin etkilerini göstermek.
- Modellerin doğruluk, genelleme ve hız açısından avantaj ve dezavantajlarını tartışmak.

1.6 Çalışmanın Yapısı

Çalışmanın yapısı aşağıdaki şekilde düzenlenmiştir.

Bölüm 1: Giriş ve literatür taraması.

Bölüm 2: Kullanılan veri seti ve ön işleme yöntemleri.

Bölüm 3: Modellerin detaylı açıklamaları ve eğitim prosedürleri.

Bölüm 4: Deneysel sonuçlar.

Bölüm 5: Tartışma.

Bölüm 6: Sonuç

2. Veri Seti ve Ön İşleme

2.1. Veri Seti Kaynağı

Bu çalışmada kullanılan veri seti, Potato Leaf Disease Dataset adlı açık kaynaklı bir veri setinden alınmıştır. Veri setinin orijinal kaynağı, Kaggle platformunda yayınlanmaktadır. Veri seti, patates yapraklarında görülen hastalıkların sınıflandırılmasına yönelik yüksek çözünürlüklü görüntüler içermektedir. Veri setinin örnek görüntüleri şekil 1'de gösterilmiştir.

Şekil 1. Veri setinin örnek gösterimi.

2.2. Veri Seti İçeriği

Bu veri seti, toplam 1500 görüntüden oluşmaktadır ve her biri belirli bir sınıfa ait yaprak görüntülerini temsil etmektedir. Görüntüler, üç ana sınıfa ayrılmıştır.

Erken Yanıklık (Early Blight):Alternaria solani adlı bir patojenin neden olduğu bir hastalıktır. Görüntülerde genellikle düzensiz kahverengi lekeler ve yaprak kenarlarında deformasyonlar görülmektedir (Van der Waals ve ark., 2001).

Geç Yanıklık (Late Blight): Phytophthora infestans adlı bir patojenin neden olduğu ciddi bir hastalıktır. Yapraklarda büyük, koyu lekeler ve küf benzeri izler bulunmaktadır (Bista ve Adhikari,2023).

Sağlıklı (Healthy): Hiçbir hastalık belirtisi olmayan, sağlıklı patates yapraklarını temsil eder. Görüntüler, düzgün yüzey dokusuna ve homojen yeşil renge sahiptir (Gold ve ark., 2020).

Her sınıfta 500 görüntü bulunmaktadır, bu da veri setinin dengeli bir yapıya sahip olduğunu göstermektedir. Bu sınıf dengesi, modellerin öğrenme sürecinde sınıf yanlılığını (class imbalance) engellemeye yardımcı olmaktadır. Veri seti dağılımı Tablo 1'de gösterilmiştir.

Tablo 1. Veri setinin dağılımı.

Sınıf	**Görüntü Sayısı**	**Oran (%)**
Erken Yanıklık	500	%33.33
Geç Yanıklık	500	%33.33
Sağlıklı	500	%33.33
Toplam	1500	%100

2.3. Veri setinin bölünmesi

Modelin eğitimi, doğrulanması ve test edilmesi amacıyla veri seti aşağıdaki şekilde bölünmüştür:

- Eğitim (Train) Veri Seti: 1050 görüntü (%70)
- Doğrulama (Validation) Veri Seti: 225 görüntü (%15)
- Test (Test) Veri Seti: 225 görüntü (%15)

Bu bölünme, modelin genelleme yeteneğini değerlendirmek ve ezberlemeyi (overfitting) önlemek için ideal bir dağılım sunmaktadır.

2.4. Ön İşleme Yöntemleri

Çalışmada, öğrenmenin daha verimli olabilmesi için derin öğrenme modelinden önce, veri seti üzerinde çeşitli ön işleme adımları uygulanmıştır:

2.4.1.Görüntü boyutlandırma

Görüntüler kullanılacak modelin giriş boyutuna uygun hale getirilmelidir. Bu yüzden tüm görüntüler VGG16 için 224x224, ViT için ise 384x384 piksel boyutuna yeniden boyutlandırılmıştır.

2.4.2.Normalizasyon

Derin öğrenme modellerinin daha verimli kullanılabilmesi için normalizasyon işlemi yapılmaktadır. Görüntü piksel değerleri [0, 1] aralığına ölçeklenmiştir. VGG16 ve ViT modellerinin gereksinimlerine uygun olarak RGB kanalları için ortalama ve standart sapma değerleri belirlenmiştir. Bu değerler VGG16 için ortalama (Mean): (0.485, 0.456, 0.406), standart sapma (Std): (0.229, 0.224, 0.225) değerleridir. ViT için ise ortalama (Mean)=(0.5, 0.5, 0.5),), standart sapma (Std): (0.5, 0.5, 0.5)) şeklindedir.

2.4.3.Etiketleme (Labeling)

Derin öğrenme modeli ile sınıflandırma yapabilmek için her bir görüntü etiketlenmelidir. Çalışmada, her görüntü, ait olduğu sınıf etiketiyle (0: Erken Yanıklık, 1: Geç Yanıklık, 2: Sağlıklı) ilişkilendirilmiştir.

2.5. Veri Setinin Avantajları ve Zorlukları

Veri setinin dengeli sınıf dağılımı, modellerin adil bir şekilde eğitilmesini sağlamaktadır. Ayrıca görüntülerin yüksek çözünürlüğü, hastalık belirtilerinin daha iyi tanımlanmasına olanak tanımaktadır.

Bunların yanında veri setinin boyutu (1500 görüntü), daha büyük modeller için genelleme kapasitesini sınırlandırabilmektedir.

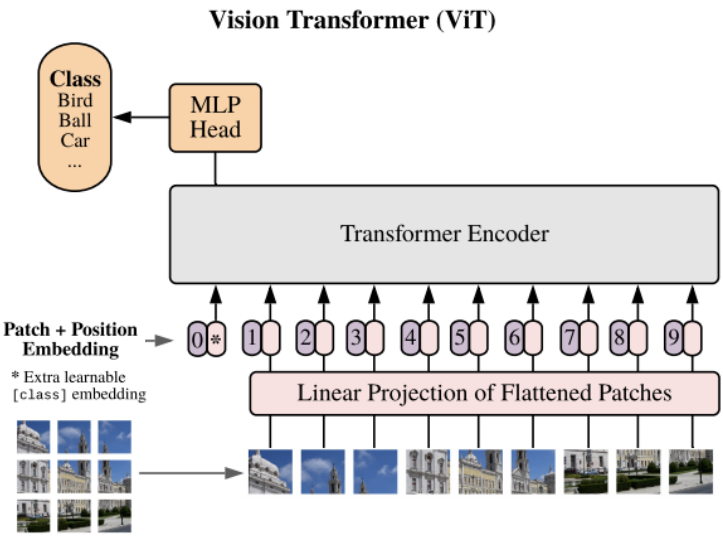
Görüntülerin farklı ışık ve ortam koşullarında çekilmiş olması, modelin öğrenme sürecini etkilemektedir (Karadeniz, Başaran ve Celik., 2023).

3. Derin Öğrenme Modelleri

Bu çalışmada, patates hastalıklarının yaprak görüntülerinden sınıflandırılması için iki farklı derin öğrenme modeli, Vision Transformer (ViT) ve VGG16 kullanılmıştır. Modellerin mimari yapıları, eğitim süreçleri ve avantajları aşağıda açıklanmıştır.

3.1. Vision Transformer (ViT) Model Yapısı

Vision Transformer (ViT), özellikle büyük veri setleri üzerinde etkili bir şekilde çalışan transformer tabanlı bir görüntü sınıflandırma modelidir. Bu model, görüntüleri sabit boyutlu parçalara (patch) bölmekte ve her bir parçayı bir kelime gibi işlem yaparak öğrenmektedir. Çalışmada kullanılan model, önceden eğitilmiş "google/vit-large-patch32-384" modelidir. Bu model, 384x384 boyutunda giriş görüntüleriyle çalışmaktadır. Kullanılan her yama (patch) 32x32 piksel boyutundadır. Vit model mimarisi şekil 2’de gösterilmiştir (Karadeniz, 2024).



Şekil 2. ViT mimarisi.

ViT derin öğrenme modelinin en belirgin özellikleri aşağıda verilmiştir.

- Transformer tabanlı encoder yapısı.
- Görüntülerin geniş bağlam bilgilerini yakalamada etkili.
- Yüksek sayıda parametre ile öğrenme kapasitesi oldukça fazladır.

3.1.1. Eğitim ayrıntıları

ViT modelinin eğitim sürecinde transfer öğrenme yaklaşımı uygulanmıştır. Model, önceden eğitilmiş ağırlıkları kullanarak başlangıç yapmış ve küçük bir öğrenme oranıyla yeniden eğitilmiştir. Eğitimde kullanılan hiperparametreler aşağıdaki gibidir.

- Öğrenme Oranı (Learning Rate): 1e-5
- Epoch Sayısı: 10
- Eğitim Batch Boyutu: 16

- Doğrulama Batch Boyutu: 16

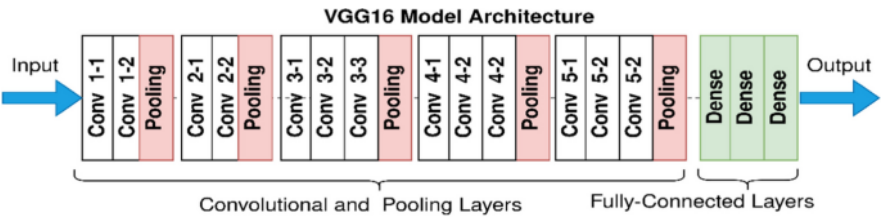
3.1.2. ViT modelinin avantajları

ViT modeli, transformer tabanlı yapısı sayesinde, özellikle büyük ölçekli veri setlerinde yüksek doğruluk oranları elde edebilmektedir. Uzun mesafeli bağımlılıkları, öğrenme yeteneğini arttırmakta ve daha karmaşık özelliklerin yakalanmasına olanak sağlamaktadır. Önceden eğitilmiş modellerin kullanımı, küçük veri setlerinde bile güçlü performans sergilemesini yardımcı olmaktadır (Zhang ve ark., 2023).

3.2. VGG16 Model Yapısı

VGG16, geleneksel Convolutional Neural Network (CNN) tabanlı bir modeldir ve genellikle transfer öğrenim için yaygın olarak kullanılmaktadır. Model, sabit boyutlu (224x224) giriş görüntülerine uygun olacak şekilde 13 evrişim katmanı ve 3 tam bağlı katman olamak üzere toplan 16 katman içerirmektedir.

VGG16 mimarisinin genel yapısı şekil 3'de gösterilmiştir (Karadeniz ve ark., 2024).



Şekil 3. VGG16 mimarisi.

Bu çalışmada, önceden eğitilmiş VGG16 modeli kullanılmıştır. Modelin son tam bağlı katmanı sınıf sayısına göre şu şekilde özelleştirilmiştir:

- model = models.vgg16(pretrained=True)
- model.classifier[3] = nn.Linear(4096, num_classes)

Bu yapılandırma, modelin çıkış boyutunun, mevcut veri setindeki sınıf sayısına (3: Erken Yanıklık, Geç Yanıklık, Sağlıklı) uygun hale getirilmesini sağlamıştır.

3.2.1 Eğitim Ayrıntıları

VGG16 modelinin eğitim sürecinde aşağıdaki hiperparametreler kullanılmıştır:

- Öğrenme Oranı (Learning Rate): 1e-4
- Epoch Sayısı: 10
- Batch Boyutu (Batch Size): 32
- Loss Fonksiyonu: CrossEntropyLoss
- Optimizasyon Algoritması: SGD (Stochastic Gradient Descent)

3.2.2. Avantajları

VGG16, daha basit ve anlaşılır bir mimariye sahiptir, bu da özellikle yeni başlayanlar için kolaylık sağlamaktadır. Önceden eğitilmiş ağırlıklar, küçük ve orta ölçekli veri setlerinde etkili bir şekilde transfer öğrenme yapılmasına olanak tanımaktadır (Tammina, 2019).

Daha az karmaşık bir yapıya sahip olduğu için eğitim süresi ViT'ye göre genellikle daha kısadır.

Çalışmada, ViT ve VGG16 modellerinde kullanılan hiperparametreler tablo 2'de gösterilmiştir.

Tablo 2. Modellerin hiperparametreleri.

Özellik	**Vision Transformer (ViT)**	**VGG16**
Giriş Boyutu	384x384	224x224
Model Yapısı	Transformer tabanlı	CNN tabanlı
Öğrenme Oranı	1e-5	1e-4
Epoch Sayısı	10	10
Batch Boyutu	16	32
Optimizasyon Algoritması	Adam	SGD

4. Deneysel Sonuçlar

Bu bölümde, Vision Transformer (ViT) ve VGG16 modellerinin eğitim, doğrulama ve test veri setleri üzerindeki performans sonuçları detaylı bir şekilde analiz edilmiştir. Modellerin doğruluk, kayıp değerleri ve diğer metrikler açısından karşılaştırmaları yapılmıştır.

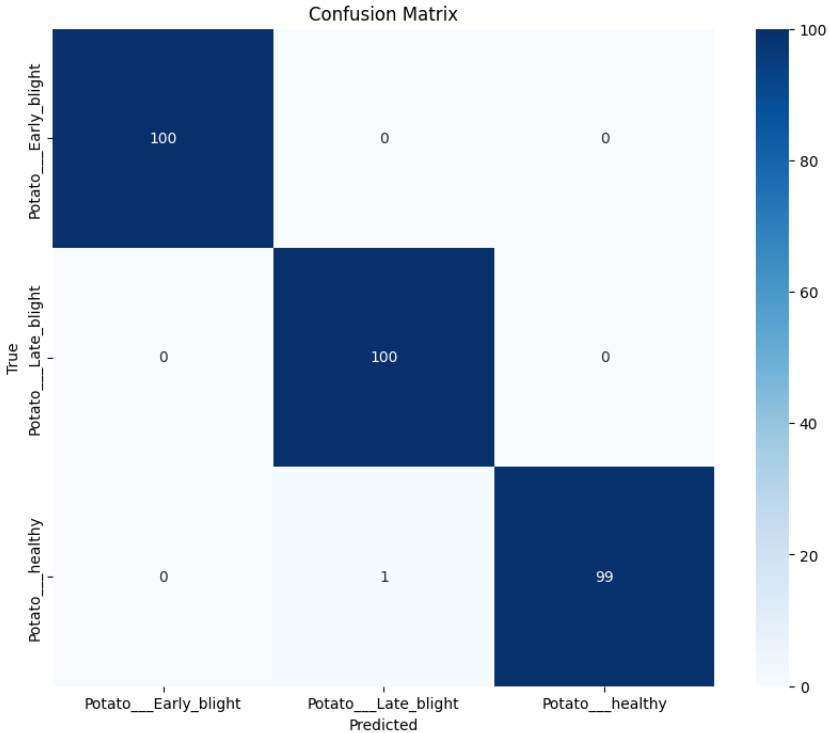
4.1. Vision Transformer (ViT) Test Sonuçları

Vision Transformer modeli, patates yaprak görüntülerinden hastalıkların sınıflandırılmasında çok iyi bir performans sergilemiştir. ViT ile yapılan deneysel test sonuçları Tablo 3'de gösterilmiştir.

Tablo 3. ViT ile test sonuçları.

Metrik	**Sonuç**
Doğrulama Doğruluğu (Validation Accuracy)	1.0
Test Kaybı (Loss)	0.0108
Doğruluk (Accuracy)	0.9967
Hassasiyet (Precision)	0.9967
Duyarlılık (Recall)	0.9967
F1 Skoru	0.9967
İşlenen Görüntüler/Saniye	26.195

Bu sonuçlar, ViT modelinin test veri setindeki sınıflandırma başarısının çok yüksek olduğunu göstermektedir. Model, düşük kayıp değeriyle birlikte hem doğruluk hem de diğer metriklerde %99.67 oranında performans göstermiştir. ViT ile yapılan deneysel çalışmanın karmaşıklık matrisi şekilde 4'de verilmiştir.



Şekil 4. ViT modeli ile yapılan test sonuçlarının karmaşıklık matrisi.

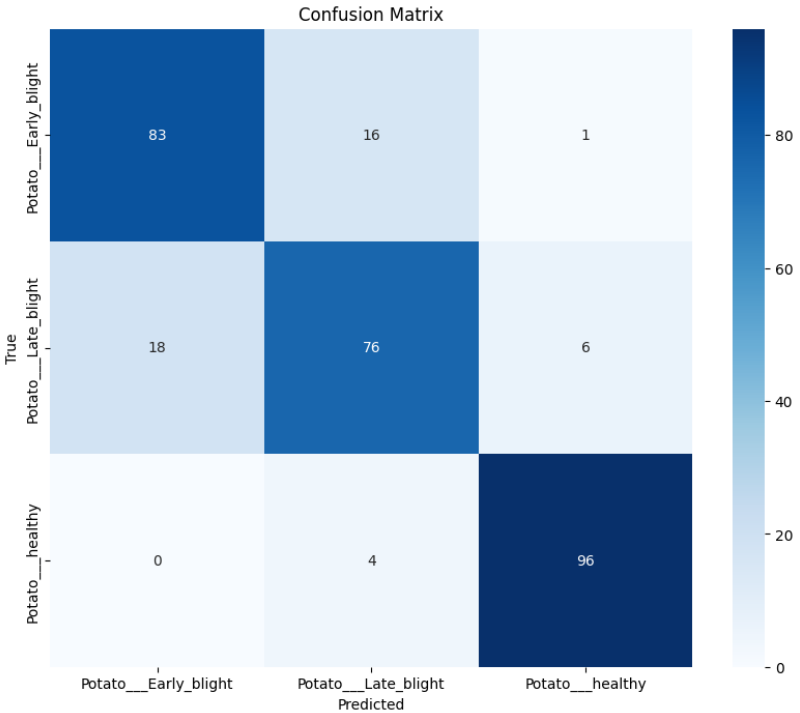
4.2. VGG16 Test Sonuçları

VGG16 modeli, daha geleneksel bir CNN mimarisi olmasına rağmen patates yaprak görüntülerinden hastalıkların sınıflandırılmasında iyi bir performans sergilemiştir. Eğitim, doğrulama ve test sonuçları tablo 4'de gösterilmiştir.

Tablo 4. VGG16 ile test sonuçları.

Metrik	**Sonuç**
Doğrulama Doğruluğu (Validation Accuracy)	0.8767
Doğruluk (Accuracy)	0.8500
Hassasiyet (Precision)	0.8485
Duyarlılık (Recall)	0.8500
F1 Skoru	0.8491

Bu sonuçlar, VGG16 modelinin küçük bir veri seti üzerinde kabul edilebilir bir performans sergilediğini göstermektedir. Model, özellikle doğrulama doğruluğunda (87.67%) yüksek bir başarı elde etmiştir. VGG16 ile yapılan deneysel test sonuçlarının karmaşıklık matrisi şekil 5'de verilmiştir.



Şekil 5. VGG16 modeli ile yapılan test sonuçlarının karmaşıklık matrisi.

4.3. Derin öğrenme modellerinin Performans Karşılaştırması

Tablo 5'de Vision Transformer (ViT) ve VGG16 modellerinin test sonuçlarını karşılaştırmaktadır:

Tablo 5. Model test sonuçlarnın karşılaştırma tablosuı.

Metrik	ViT	VGG16
Validation Accuracy	1.0	0.8767
Test Loss	0.0108	0.6574
Test Accuracy	0.9967	0.8500
Test Precision	0.9967	0.8485
Test Recall	0.9967	0.8500
Test F1 Score	0.9967	0.8491
Test Runtime (s)	11.4527	32.528

4.3. Deneysel test sonuçlarının analizi

4.3.1.Model Performansı

Vision Transformer modeli, test doğruluğu (%99.67), F1 skoru (%99.67) ve diğer metriklerde VGG16 modeline kıyasla önemli ölçüde daha yüksek performans göstermiştir.

VGG16 modeli ise küçük veri setlerinde etkili bir performans sunmuş, ancak ViT'nin başarısına ulaşamamıştır.

4.3.2. Overfitting Sorunu

ViT modeli, doğrulama doğruluğu ile test doğruluğu arasındaki tutarlılık nedeniyle ezberleme (overfitting) yapmamıştır.

VGG16 modelinde, eğitim doğruluğu ile test doğruluğu arasındaki fark, modelin biraz ezberleme (overfitting) yapabileceğini işaret etmektedir.

4.3.3. Hız ve Verimlilik

ViT modeli test işlemlerinde oldukça hızlıdır ve saniyede 26 görüntü işleyebilmiştir. Bu da tarımsal uygulamalarda gerçek zamanlı kullanım için uygundur.

4.4.4. Uygulama Alanları

Vision Transformer, büyük veri setlerinde daha etkili olabilirken, VGG16 modeli küçük veri setlerinde hızlı ve etkili bir başlangıç modeli olarak tercih edilebilir.

Bu sonuçlar, Vision Transformer modelinin patates yaprak görüntülerinden hastalıklarının teşhisinde üstün bir performans sergilediğini ve tarımsal yapay zeka uygulamaları için uygun bir çözüm sunduğunu göstermektedir. Ancak, uygulama koşulları ve donanım kısıtlamalarına göre VGG16'nın da bir alternatif olarak kullanılabileceği sonucuna varılmıştır.

5. Tartışma

5.1. Modellerin Performans Farklılıkları

Bu çalışmada kullanılan iki model olan Vision Transformer (ViT) ve VGG16, patates yaprak görüntülerinden hastalıklarının sınıflandırılmasında farklı performans seviyeleri sergilemiştir. ViT modeli, hem doğrulama hem de test veri setlerinde %99.67 doğruluk oranıyla üstün bir performans göstermiştir. Transformer tabanlı mimarisi, uzun menzilli bağımlılıkları öğrenme yeteneği sayesinde daha geniş bağlam bilgilerini yakalayarak karmaşık özelliklerin çıkarılmasını mümkün kılmaktadır. Ancak, VGG16 modeli %85 doğruluk oranıyla kabul edilebilir bir performans sergilemiş olsa da, ViT kadar yüksek bir başarı sağlayamamıştır. VGG16'nın daha klasik bir CNN mimarisi, özellikle küçük veri setlerinde genelleme kapasitesini sınırlayabilmektedir.

5.2. Veri Setinin Boyutunun ve Çeşitliliğinin Etkileri

Çalışmada kullanılan veri seti, yalnızca 1500 görüntüden oluştuğu için oldukça sınırlıdır. Bu durum, özellikle büyük veri setleri için optimize edilmiş ViT modeli için bir dezavantaj

oluşturabilecekken, transfer öğrenme yaklaşımı bu sorunun üstesinden gelmiştir. VGG16 gibi daha az karmaşık modeller, küçük veri setlerinde iyi bir başlangıç performansı sergilese de, veri çeşitliliğinin artırılması modelin genelleme kapasitesini daha fazla artırabilirdi. Veri artırma tekniklerinin VGG16 üzerinde uygulanmış olması, modelin doğrulama doğruluğunu artırmış ancak test doğruluğu açısından ViT ile rekabet edememiştir.

5.3. Eğitim Süresi ve Donanım Gereksinimleri

ViT modeli, daha karmaşık bir mimariye sahip olması nedeniyle eğitim süresi açısından daha fazla kaynak ve zaman gerektirmiştir. Ancak, test sürecinde saniyede 26 görüntü işleyebilmesi, ViT'nin gerçek zamanlı uygulamalarda daha etkili olabileceğini göstermektedir. VGG16, daha hızlı eğitim süresi ve daha düşük donanım gereksinimleriyle dikkat çekerken, test süresi ölçülmemiş olmasına rağmen daha uzun olduğu tahmin edilmektedir. Bu nedenle, uygulama koşullarına bağlı olarak VGG16 da bir alternatif olarak düşünülebilir.

5.4. Ezberleme (Overfitting) Sorunu

ViT modeli, doğrulama doğruluğu (%100) ile test doğruluğu (%99.67) arasında tutarlı bir performans sergilemiş ve ezberleme (overfitting) belirtileri göstermemiştir. Ancak, VGG16 modelinde eğitim doğruluğu (%72.44) ile doğrulama doğruluğu (%87.67) arasındaki fark, modelin overfitting eğilimi taşıyabileceğini işaret etmektedir. Bu durum, VGG16'nın genelleme kapasitesinin ViT'ye kıyasla daha düşük olduğunu ortaya koymaktadır.

6. Sonuç

Bu çalışmada, Vision Transformer (ViT) ve VGG16 modelleri, patates yaprak görüntülerinden hastalıklarının sınıflandırılması için karşılaştırılmıştır. Sonuçlar, ViT'nin %99.67 doğruluk oranıyla üstün bir performans sergilediğini göstermiştir. Transformer tabanlı yapısı, uzun menzilli bağımlılıkları öğrenme yeteneği sayesinde daha karmaşık özellikleri anlamada avantaj

sağlamıştır. Öte yandan, VGG16 modeli %85 doğruluk oranıyla küçük veri setlerinde kabul edilebilir bir performans sunmuştur.

Tarımsal teşhis uygulamaları için, veri seti boyutu ve çeşitliliği, model seçimi kadar önemlidir. ViT, büyük veri setlerinde daha etkili bir çözüm sunarken, VGG16 daha az kaynak gereksinimi olan uygulamalarda tercih edilebilir. Bu çalışma, tarım sektöründe yapay zeka tabanlı çözümlerin etkinliğini artırmak ve tarımsal sürdürülebilirliği desteklemek için önemli bir referans niteliğindedir.

Bu çalışmada kullanılan veri seti boyutunun sınırlı olması, modellerin potansiyel performanslarını tam olarak değerlendirmeyi zorlaştırmıştır. Gelecekteki çalışmalar, daha büyük ve çeşitli tarımsal veri setleri üzerinde yapılmalıdır. Ayrıca, ViT ve VGG16'nın hibrit bir model olarak birleştirilmesi, hem yüksek doğruluk hem de hızlı eğitim süresi sağlayabilecek yeni bir yaklaşım sunabilir. Farklı derin öğrenme modellerinin de karşılaştırılması, tarımsal teşhis sistemlerinde en uygun modelin belirlenmesine yardımcı olacaktır.

Kaynaklar

Adhikari, S. (2024). Advancements in Agricultural Technology: Vision Transformer-Based Potato Leaf Disease Classification. Journal of Soft Computing Paradigm, 6(2), 169–185.

Ale, L., Sheta, A., Li, L., Wang, Y., & Zhang, N. (2019). Deep learning based plant disease detection for smart agriculture. 2019 IEEE Globecom Workshops (GC Wkshps), 1–6.

Asif, M. K. R., Rahman, M. A., & Hena, M. H. (2020). CNN based disease detection approach on potato leaves. 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), 428–432.

Bista, S., & Adhikari, A. (n.d.). A COMPREHENSIVE STRATEGY FOR LATE BLIGHT MANAGEMENT IN POTATO AND TOMATO.

Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. ArXiv Preprint ArXiv:2010.11929.

Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145, 311–318.

Fiers, M., Edel-Hermann, V., Chatot, C., Le Hingrat, Y., Alabouvette, C., & Steinberg, C. (2012). Potato soil-borne diseases. A review. Agronomy for Sustainable Development, 32(1), 93–132.

Gold, K. M., Townsend, P. A., Chlus, A., Herrmann, I., Couture, J. J., Larson, E. R., & Gevens, A. J. (2020). Hyperspectral measurements enable pre-symptomatic detection and differentiation

of contrasting physiological effects of late blight and early blight in potato. Remote Sensing, 12(2), 286.

Haidari, A., & Kumar, A. (2023). Potato Leaf Disease Detection Using the Convolutional Neural Network. 2023 14th International Conference on Electrical and Electronics Engineering (ELECO), 1–5.

Husna, A. (2024). Characterization of Potato Leaf Disease by Digital Image Processing Technique. Journal of Agriculture, Food and Environment (JAFE)| ISSN (Online Version): 2708-5694, 5(2), 13–18.

Karadeniz, A. T. (n.d.). Automatic Classification of Melanoma Skin Cancer Images with Vision Transform Model and Transfer Learning. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 13(3), 844–850.

Karadeniz, A. T., Başaran, E., & Çelik, Y. (2024). Classification of walnut dataset by selecting CNN features with whale optimization algorithm. Multimedia Tools and Applications, 1–16.

Karadeniz, A. T., Başaran, E., & Celık, Y. (2023). Identification of walnut variety from the leaves using deep learning algorithms. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 12(2), 531–543.

Karadeniz, A. T., Çelik, Y., & Başaran, E. (2023). Classification of walnut varieties obtained from walnut leaf images by the recommended residual block based CNN model. European Food Research and Technology, 249(3), 727–738.

Li, L., Zhang, S., & Wang, B. (2021). Plant disease detection and classification by deep learning—a review. IEEE Access, 9, 56683–56698.

Nawaz, M., Nazir, T., Baili, J., Khan, M. A., Kim, Y. J., & Cha, J.-H. (2023). CXray-EffDet: chest disease detection and classification from X-ray images using the EfficientDet model. Diagnostics, 13(2), 248.

Pasalkar, J., Gorde, G., More, C., Memane, S., & Gaikwad, V. (2023). Potato Leaf disease detection using machine learning. Current Agriculture Research Journal, 11(3).

Rozaqi, A. J., Arief, M. R., & Sunyoto, A. (2021). Implementation of transfer learning in the convolutional neural network algorithm for identification of potato leaf disease. Procedia of Engineering and Life Science, 1(1).

Sharma, M., & Sharma, N. (2024). PotatoLeafGuard: A Novel Hybrid CNN Architecture for Advanced Detection of Potato Leaf Diseases. 2024 International Conference on Emerging Innovations and Advanced Computing (INNOCOMP), 312–317.

Sholihati, R. A., Sulistijono, I. A., Risnumawan, A., & Kusumawati, E. (2020). Potato leaf disease classification using deep learning approach. 2020 International Electronics Symposium (IES), 392–397.

Singh, A. K., Sreenivasu, S. V. N., Mahalaxmi, U., Sharma, H., Patil, D. D., & Asenso, E. (2022). [Retracted] Hybrid Feature-Based Disease Detection in Plant Leaf Using Convolutional Neural

Network, Bayesian Optimized SVM, and Random Forest Classifier. Journal of Food Quality, 2022(1), 2845320.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. Computational Intelligence and Neuroscience, 2016(1), 3289801.

Tammina, S. (2019). Transfer learning using vgg-16 with deep convolutional neural network for classifying images. International Journal of Scientific and Research Publications (IJSRP), 9(10), 143–150.

Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture, 161, 272–279.

Van der Waals, J. E., Korsten, L., & Aveling, T. A. S. (2001). A review of early blight of potato. African Plant Protection, 7(2), 91–102.

Wang, Y., Huang, R., Song, S., Huang, Z., & Huang, G. (2021). Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. Advances in Neural Information Processing Systems, 34, 11960–11973.

Zhang, L., Wu, J., Zhang, K., Wang, Z., Yan, X., Liu, P., Wang, Q., Fan, L., Yao, J., & Yang, Y. (2023). Diagnosis of pumping machine working conditions based on transfer learning and ViT model. Geoenergy Science and Engineering, 226, 211729.

BÖLÜM VII

Brats20 veri seti ile beyin tümörü segmentasyonunda Segnet ve MambaUnet'in derin öğrenme performanslarının karşılaştırılması

Rabia UYAR[1]

GİRİŞ

Beyin tümörü, insan sağlığı üzerinde önemli etkileri olan ve erken teşhis edilip tedavi yapılmazsa ölümle sonuçlanan kritik hastalıklardan biridir (Shiri vd., 2024). Beyin tümörlerinin hastaların tedavi planlarının belirlenmesi, yaşam kalitelerini artırılması ve yaşam sürelerinin uzatılabilmesi için, doğru ve hızlı bir şekilde tespit edilmesi büyük önem arz etmektedir (Pérez Cáceres vd., 2024). Bu kapsamda tümörlerin tipi, büyüklüğü ve konumunun hızlı bir şekilde belirlenmesi tedavi sürecinde kritik öneme sahiptir ve güvenilir segmentasyon süreci, beyin tümörü uzmanları için vazgeçilmezdir (S. K. Zhou vd., 2021).

Manuel segmentasyon yöntemleri, insan gücü gerektiren süreçler ve geleneksel yöntemler, subjektif ve zaman alıcı olabilmektedir. Aynı

Arş Gör., Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Kütahya/TÜRKİYE, Orcid: 0000-0002-6565-0560, rabia.uyar@dpu.edu.tr

zamanda bu yöntemler, uzmanlık ve deneyim gerektirdiği için, hata payını da artırmaktadır (Bouchakwa vd., 2020). Ayrıca manuel segmentasyonun tekrarlanabilirliği düşük olup, aynı tümör görüntüsü farklı nöroloji uzmanı tarafından segment edilip değerlendirildiğinde farklı sonuçlar ortaya çıkabilmektedir. Bu sınırlamalar ise, otomatik segmentasyon yöntemleri ve tıbbi görüntü analizi alanında daha etkili ve insan gücü gerektirmeyen otomatik yöntemlerin geliştirilmesini zorunlu kılmaktadır (Q. Zhou vd., 2023).

Otomatik segmentasyon ve görüntü analizi yöntemlerinin geliştirilip iyileştirilmesi, tıp alanında büyük öneme sahiptir (Abdelmalek vd., t.y.; Bohlender vd., 2021; Kaur vd., 2022; T. Zhou vd., 2023). Çünkü bu otomatik yöntemler, insan faktöründen kaynaklanan subjektifliği ve hataları azaltarak, daha hızlı ve tekrarlandığı takdirde daha tutarlı sonuçlar üretebilmektedirler (Noor & Ige, 2024; Rahmani vd., 2024). Aynı zamanda bu teknolojiler, büyük veri kümelerini işleyebilmekte, bu sayede ise daha hızlı teşhis ve tedaviye olanak tanımaktadır (Lundervold & Lundervold, 2019; Siddique vd., 2021).

Günümüzde, makine öğrenmesi ve derin öğrenme yöntemleri, tıbbi görüntüleme alanında önemli ilerlemeler kaydetmiştir (Abdusalomov vd., 2023; Jimenez-Mesa vd., 2023). Derin öğrenmede kullanılan konvolüsyonel sinir ağları (CNN'ler), büyük ve düzensiz veri setlerini işleyip değerlendirebilme yeteneklerine sahiptirler. Bu teknolojiler, tıbbi görüntülerin otomatik analizi ve eğitim süreci boyunca verilerden öğrenerek, yüksek doğrulukta segmentasyon yapabilme kapasitesine sahiptir (Mohd Adam vd., 2020; Nam & Pak, 2024).

Bu çalışmada, MambaUnet ve SegNet olmak üzere iki farklı derin öğrenme mimarisi kullanılarak, beyin tümörü segmentasyonu yapılmış ve sonuçları karşılaştırılmaktadır. MambaUnet, Unet'in geliştirilmiş versiyonu olup daha karmaşık yapısı bulunmaktadır. SegNet'in ise görüntü segmentasyonunda başarılı sonuçlar sunduğu görülmektedir (Liu vd., 2023; Ming vd., 2022; Mohd Adam vd.,

2020). Kısaca bu iki modele bakıldığında, her biri farklı mimari özelliklere, avantajlara ve dezavantajlara sahiptir (Chan vd., 2023; Martinez-Murcia vd., 2021; Toğaçar, 2021).

MambaUNet mimari yapısı, Unet mimari yapısına göre daha derin katmanlar ve daha fazla parametre kullanmaktadır ve özellikle karmaşık segmentasyon problemlerinde üstün performans göstermektedir (Aggarwal vd., 2021; Altaf vd., 2019). Model daha fazla hesaplama işlemi gerektirmesine rağmen yüksek doğruluk ve detaylı segmentasyon analizi sunmaktadır.

SegNet mimari yapısı, giriş görüntüsünü düşük çözünürlüklü geçici bir temsile dönüştürür ve ardından bu temsili Encoder ve dekoder yapısı sayesinde tekrar yüksek çözünürlüklü bir çıktıya dönüştürür. Bu dönüşüm süreci, görüntü verisindeki önemli özelliklerin korunmasını sağlar ve segmentasyonun doğruluk değerini artırır (Eminaga vd., 2023; Soomro vd., 2023). SegNet'in bu basit yapısı hesaplama maliyetini düşürür ve hızlı sonuçlar elde etmeye olanak tanır.

Brats2020 veri seti, beyin tümörü segmentasyonunda yaygın olarak kullanılan farklı modalitelerde (T1, T1c, T2, FLAIR gibi) manyetik rezonans görüntülerini içeren kapsamlı bir veri setidir. Veri seti farklı tümör türleri ve özellikleri içermektedir. Bu sayede model eğitimi ve değerlendirilmesi için ideal kapsam ve büyüklüktedir. Veri setinin zengin veri çeşitliliği, modellerin genellemesine olanak sağlar ve gerçek klinik senaryolarda kullanılabilirliğini artırmaktadır.

Brats2020 veri seti kullanılarak gerçeklenen bu çalışmada ise MambaUnet ve SegNet mimarilerinin performansları (Ortalama dice skoru, ortalama IoU skoru, doğruluk, kesinlik, duyarlılık, F1-skoru, özgüllük değerleri) karşılaştırılacak ve her bir mimarinin güçlü ve zayıf yönleri belirlenip değerlendirme yapılacaktır. Performans değerlendirmesi için Dice benzerlik katsayısı, Jaccard indeksi ve hassasiyet gibi yaygın metrikler kullanılacaktır. Kullanılan bu metrikler görüntü segmentasyonunun doğruluğunu ve modellerin performansını objektif şekilde ölçmek için kullanılmaktadır. Ve bu

sayede, modellerin gerçek klinik uygulamalarındaki performansları değerlendirilecektir.

Çalışmanın amacı, beyin tümörü segmentasyonu için en uygun derin öğrenme mimarisini belirlemek ve bu mimarinin tıp alanındaki çalışmalarda kullanılabilirliğini değerlendirmektir. Elde edilen sonuçların, gelecek araştırmalara ve tıp alanındaki görüntü segmentasyonu uygulamalarında, önemli katkılar sağlayacağı öngörülmektedir.

MATERYAL VE YÖNTEM

1. MambaUnet

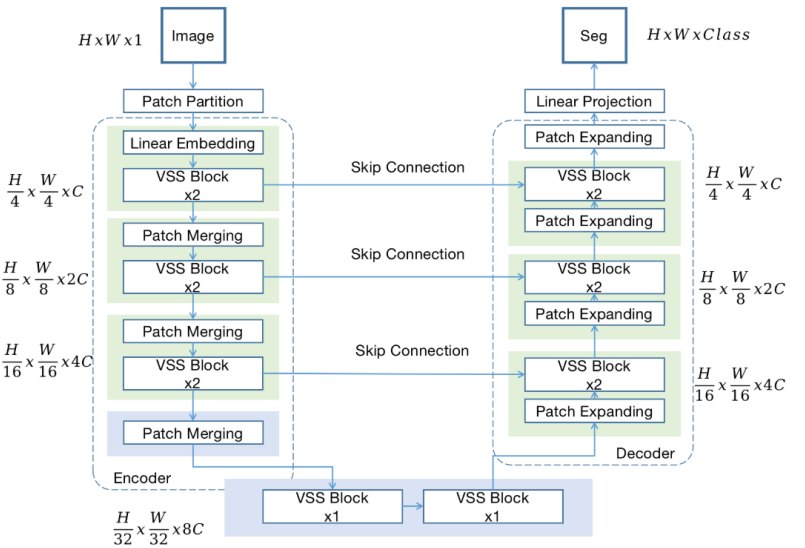
MambaUnet, giriş görüntüsünü HxWx1 boyutunda alır ve bu görüntüyü küçük parçalara bölen bir "Patch Partition" katmanı ile başlar. Her bir parça, daha sonra "Linear Embedding" katmanında daha yüksek boyutlu vektörlere dönüştürülür. Encoder aşamasında, her bir aşamada "VSS Block" modülleri kullanılarak özellik çıkarımı yapılır ve bu modüller ardışık olarak yerleştirilmiştir. Özellik haritaları belirli aşamalarda "Patch Merging" işlemiyle birleştirilir, bu sayede boyutları küçültülür ve bilgi yoğunluğu artırılır.

Decoder aşamasında, encoder tarafından sıkıştırılmış olan düşük boyutlu temsilcikler tekrar genişletilir. Bu işlem, "Patch Expanding" adı verilen katmanlar tarafından gerçekleştirilir. Decoder, encoder tarafından elde edilen bilgi ile doğrudan bağlantı kurarak daha zengin ve detaylı özellik haritaları oluşturur. "VSS Block" modülleri, decoder aşamasında da kullanılarak öğrenilen özelliklerin detaylandırılması sağlanır. Bu süreç sonunda, model giriş görüntüsüne karşılık gelen segmentasyon maskesini üretir.

1.1.MambaUnet mimari yapısı

MambaUnet mimarisi, beyin tümörü segmentasyonu gibi tıbbi görüntü işleme görevleri için geliştirilmiş bir derin öğrenme modelidir (Houssein vd., 2023). Model, U-Net benzeri encoder-decoder yapısına sahiptir ve giriş görüntülerini daha düşük boyutlu temsilciklere sıkıştırarak ve bu temsilcikleri yeniden genişleterek

çalışır. Encoder kısmında görüntüler, belirli aşamalarda alt örneklenir ve özellik haritalarına dönüştürülür. Patch Partition aşamasında giriş görüntüleri küçük parçalara ayrılır. Linear Embedding aşamasında bu parçalar daha yüksek boyutlu vektörlere dönüştürülür. VSS Block modülleri, özellik çıkarımı yaparak bilgiyi daha derinlemesine öğrenir. Patch Merging işlemi, bu bloklardan gelen bilgileri birleştirerek daha kompakt bir temsil oluşturur. Decoder kısmı, encoder tarafından üretilen düşük boyutlu temsilcikleri tekrar orijinal boyutuna genişletir. Skip Connection bağlantıları, encoder ve decoder arasında bilgi transferini sağlar. Çıkış aşamasında, model segmentasyon maskesi üretir.



Şekil 1: MambaUnet mimari yapısı

Kaynak: (Z. Wang vd., 2024)

MambaUnet'in temel yeniliklerinden biri, özellik çıkarımı ve boyut küçültme işlemlerinde kullandığı "VSS Block" modülleridir. Bu modüller, özellikle tıbbi görüntü işleme görevlerinde daha derin ve detaylı öğrenme sağlar (Peng & Wang, 2021). Model, skip

connections sayesinde encoder ve decoder arasında bilgi transferini optimize ederek daha başarılı segmentasyon sonuçları elde eder (Mahmood vd., 2023; Rodrigues vd., 2024). Ayrıca, hem encoder hem de decoder aşamalarında ardışık olarak yerleştirilen "Patch Merging" ve "Patch Expanding" katmanları, modelin bilgi yoğunluğunu verimli bir şekilde yönetmesini sağlar. Bu özellikler, MambaUnet'i tıbbi görüntü işleme görevlerinde güçlü ve etkili bir model haline getirir (Siddique vd., 2021).

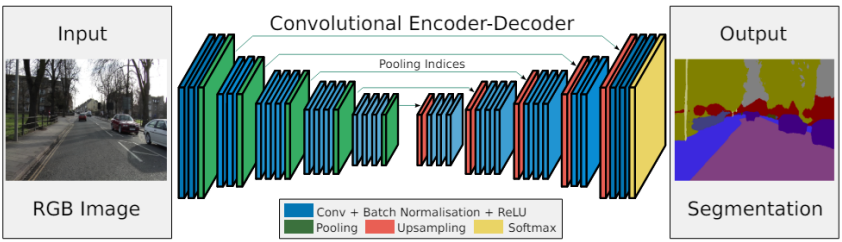
2. SegNet

SegNet, görüntü segmentasyonu için tasarlanmış bir derin öğrenme mimarisidir (Abadi vd., 2020). Temel olarak, encoder-decoder yapısını kullanarak görüntüleri piksellerine ayırır. Encoder kısmı, görüntüyü daha küçük bir boyuta indirir ve temel özellikleri çıkarır. Bu aşamada, her bir max-pooling katmanı ile görüntü boyutları yarıya indirilir. Decoder kısmı ise, encoder'dan gelen bilgiyi kullanarak görüntüyü orijinal boyutlarına geri getirir. SegNet'in önemli bir özelliği, max-pooling indekslerini encoder aşamasından decoder aşamasına aktarmasıdır. Bu sayede, konum bilgisi korunarak daha doğru bir segmentasyon sağlanır (Rahmani vd., 2024). Sonuçta, her piksel bir sınıfa atanır ve görüntü üzerinde belirli nesneler tanımlanabilir hale gelir. SegNet, özellikle yol işaretleri tanıma, tıbbi görüntü analizi ve otonom araçlar gibi uygulamalarda yaygın olarak kullanılır.

2.1.SegNet mimari yapısı

Bu mimari encoder-decoder yapısını temel alır. Giriş olarak verilen RGB görüntü, önce encoder tarafından işlenir. Encoder aşamasında, görüntü bir dizi konvolüsyon ve ReLU aktivasyon katmanlarından geçer. Bu aşamada her katman, görüntünün özelliklerini çıkarmak ve boyutunu küçültmek amacıyla kullanılır. Encoder'daki her bir konvolüsyon katmanından sonra bir max-pooling katmanı bulunur. Max-pooling katmanları, görüntünün boyutlarını yarıya indirirken, en yüksek değere sahip piksellerin konum bilgilerini de saklar. Bu konum bilgileri "pooling indices" olarak adlandırılır ve decoder aşamasında kullanılmak üzere

depolanır.



Şekil 2: SegNet mimari yapısı

Kaynak: (X. Wang vd., 2023)

Decoder aşamasında ise, encoder'dan gelen sıkıştırılmış özellik haritaları orijinal boyutlarına geri getirilir. Bu aşama, encoder'da kullanılan max-pooling işleminin tersine çevrilmesiyle gerçekleştirilir. Pooling indices, bu tersine çevirme işleminde konum bilgilerini doğru bir şekilde geri getirmek için kullanılır. Decoder'daki her bir aşamada, özellik haritaları yeniden ölçeklendirme ile genişletilir ve konvolüsyon işlemleri uygulanarak piksellerin sınıfları belirlenir. Bu sayede, segmentasyon işlemi sırasında orijinal görüntüdeki her pikselin hangi sınıfa ait olduğu belirlenir. Mimari, son olarak softmax katmanı ile her piksel için en olası sınıfı tahmin eder. Bu yapı, özellikle yol işaretleri tanıma, otonom sürüş ve tıbbi görüntü analizi gibi uygulamalarda yüksek doğruluk sağlar (Abadi vd., 2020). SegNet, derin ağların hesaplama maliyetlerini optimize ederken, yüksek doğrulukta sonuçlar elde etmek için tasarlanmıştır (Ostmeier vd., 2023).

3. Veri seti özellikleri

BRATS 2020 veri seti (Brain Tumor Segmentation 2020), beyin tümörü segmentasyonu için yaygın olarak kullanılan bir tıbbi görüntüleme veri setidir (Baştuğ Koç & Akgün, 2021). Bu veri seti, makine öğrenmesi ve derin öğrenme tabanlı modellerin geliştirilmesi, eğitilmesi ve değerlendirilmesi amacıyla

kullanılmaktadır. Özellikle Glioblastoma Multiforme (GBM) ve Düşük Dereceli Gliom (Low-Grade Glioma, LGG) gibi malign beyin tümörlerinin segmentasyonunu hedeflemektedir. Bu tümör türleri, biyolojik çeşitlilikleri ve heterojen yapıları nedeniyle segmentasyon sürecinde önemli zorluklar sunmaktadır. BRATS 2020, bu zorlukları aşabilmek için çoklu modaliteli MRG (Manyetik Rezonans Görüntüleme) verilerini kullanarak tümör bölümlerini daha hassas bir şekilde bölütlemeyi amaçlamaktadır.

BRATS 2020 veri seti, dört farklı MRG modalitesini içermektedir: T1, T1Gd, T2 ve FLAIR. T1 modalitesi standart MRG görüntülerini, T1Gd gadolinyum kontrast maddesi ile çlerini, T1Gd gadolinyum kontrast maddesi ile çekilmiş T1 ağırlıklı görüntüleri, T2 modalitesi T2 ağırlıklı görüntüleri ve FLAIR modalitesi ise sıvı baskılamalı ters dönüş (Fluid-Attenuated Inversion Recovery) görüntüleri temsil etmektedir. Bu çoklu modalite, tümör yapısının farklı bileşenlerinin (nekrotik çekirdek, kontrastla güçlenmiş aktif tümör, v.b.) daha doğru bir şekilde belirlenmesini sağlamaktadır. Veri seti, her bir hastaya ait 3B hacimsel görüntüler sunmakta olup, bu görüntüler genellikle 240x240x155 boyutundadır.

Segmentasyon etiketleri, beyin tümörü bileşenlerinin doğru bir şekilde ayrıştırılmasını sağlamak için dört farklı sınıfa ayrılmıştır: 0 (arka plan), 1 (nekrotik ve non-enhancing tümör çekirdeği), 2 (peritümöral) ve 3 (gadolinyum kontrastı ile güçlenmiş tümör alanı). Bu etiketler, her bir modalitenin belirli bölgelerindeki farklılıkları tanımlamak için kullanılır ve tümör bileşenlerinin doğru bir şekilde bölütlenmesini sağlar.

BRATS 2020 veri setinde eğitim için 369 hasta görüntüsü bulunmaktadır. Doğrulama ve test verileri ise gizli tutulmaktadır. BRATS 2020 veri seti, beyin tümörü segmentasyonu alanındaki çalışmalarda yaygın olarak kullanılır (Metlek & Çetıner, 2023). Bu veri seti, MambaUNet, SegNet. U-Net, 3D CNN ve Transformer tabanlı derin öğrenme modellerinin eğitilmesi ve test edilmesi için önemli bir kaynak sağlamaktadır. Tıbbi görüntü işleme, beyin tümörlerinin otomatik bölütlenmesi ve karar destek sistemleri gibi

alanlarda yoğun olarak kullanılmaktadır (Zeineldin vd., 2020). Bu sistemler, beyin tümörlerinin doğru teşhisi, evrelemesi ve cerrahi planlaması için klinik olarak önemli faydalar sunmaktadır.

BULGULAR VE TARTIŞMA

MambaUnet ve SegNet ağlarının performans karşılaştırmasını ele alırsak, her iki ağın da çeşitli metriklerdeki başarımları tablo 1'de sunulmaktadır.

Tablo 1: Mimarilerin performansları

ğı	Ort. Dice Skoru	Ort. IoU Skoru	Doğruluk	Kesinlik	Duyarlılık	F1-Skoru	Özgüllük
amba Unet	.7750	.7150	.9980	**.8600**	**.7948**	**3.3008**	**.9990**
eg Net	.7650	**.7050**	**.9975**	**.8720**	**.8150**	**.8430**	**.9990**

Tablo 1'deki değerler incelendiğinde MambaUnet, Ortalama Dice Skoru (0.7750), Ortalama IoU Skoru (0.7150), Doğruluk (0.9980) ve Özgüllük (0.9995) metriklerinde SegNet'e göre daha iyi sonuçlar elde etmiştir. Bu iyileştirmeler, MambaUnet'in genel segmentasyon başarısının daha yüksek olduğunu göstermektedir.

Bununla birlikte, Kesinlik (0.8600) ve Duyarlılık (0.7948) metriklerinde SegNet'e göre bir miktar düşüş gözlemlenmiştir. Özellikle, duyarlılık değerindeki bu düşüş, bazı hedef nesnelerin tespit edilememiş olabileceğini göstermektedir. Bu durum, MambaUnet'in nesneleri doğru şekilde tanıma ve tespit etme

konusunda SegNet'e göre daha fazla iyileştirme gerektirebileceğini göstermektedir.

Genel olarak, MambaUnet, 7 ölçütün 5'inde daha iyi performans sergileyerek, daha yüksek doğruluk ve daha az yanlış negatif sunmuştur. Bu, modelin pratik uygulamalarda daha güvenilir bir segmentasyon aracı olabileceğini ortaya koymaktadır. Kesinlik ve duyarlılık değerlerindeki düşüş, modelin yanlış pozitif ve yanlış negatifleri dengelemesi için ek optimizasyon gerektirdiği görülmektedir.

SONUÇ

Bu çalışmada, BRATS20 veri seti kullanılarak MambaUnet ve SegNet ağlarının beyin tümörü segmentasyonu performanslarını karşılaştırdık. Elde edilen sonuçlar, her iki ağın da çeşitli metriklerde başarılı olduğunu göstermektedir, ancak MambaUnet'in genel performansının daha yüksek olduğunu ortaya koymaktadır. Genel olarak, bu çalışmada MambaUnet'in segmentasyon performansı ve metriklerin çoğunda SegNet'ten daha iyi sonuçlar verdiği görülmektedir. Özellikle Dice ve IoU skorlarında MambaUnet'in belirgin üstünlüğü, beyin tümörü segmentasyonu görevlerinde MambaUnet'in daha tercih edilebilir bir model olduğunu ortaya koymaktadır. Elde edilen bu sonuçlar, MambaUnet'in biyomedikal görüntü segmentasyonu alanında güçlü bir araç olduğunu ve gelecekteki çalışmalar için umut vaat ettiğini göstermektedir.

KAYNAKÇA

Abadi, E., Segars, W. P., Tsui, B. M. W., Kinahan, P. E., Bottenus, N., Frangi, A. F., Maidment, A., Lo, J., & Samei, E. (2020). Virtual clinical trials in medical imaging: A review. *Journal of Medical Imaging*, *7*(04), 1. https://doi.org/10.1117/1.JMI.7.4.042805

Abdelmalek, B., Ahmed, K., & Amine, T. M. (t.y.). *A Survey on Lightweight CNN-Based Object Detection Algorithms for Platforms with Limited Computational Resources*.

Abdusalomov, A. B., Mukhiddinov, M., & Whangbo, T. K. (2023). Brain Tumor Detection Based on Deep Learning Approaches and Magnetic Resonance Imaging. *Cancers*, *15*(16), 4172. https://doi.org/10.3390/cancers15164172

Aggarwal, R., Sounderajah, V., Martin, G., Ting, D. S. W., Karthikesalingam, A., King, D., Ashrafian, H., & Darzi, A. (2021). Diagnostic accuracy of deep learning in medical imaging: A systematic review and meta-analysis. *Npj Digital Medicine*, *4*(1), 65. https://doi.org/10.1038/s41746-021-00438-z

Altaf, F., Islam, S. M. S., Akhtar, N., & Janjua, N. K. (2019). Going Deep in Medical Image Analysis: Concepts, Methods, Challenges, and Future Directions. *IEEE Access*, *7*, 99540-99572. https://doi.org/10.1109/ACCESS.2019.2929365

Baştuğ Koç, A., & Akgün, D. (2021). U-net Mimarileri ile Glioma Tümör Segmentasyonu Üzerine Bir Literatür Çalışması. *European Journal of Science and Technology*. https://doi.org/10.31590/ejosat.959590

Bohlender, S., Oksuz, I., & Mukhopadhyay, A. (2021). *A survey on shape-constraint deep learning for medical image segmentation* (No. arXiv:2101.07721). arXiv. https://doi.org/10.48550/arXiv.2101.07721

Bouchakwa, M., Ayadi, Y., & Amous, I. (2020). A review on visual content-based and users' tags-based image annotation: Methods and techniques. *Multimedia Tools and Applications*, *79*(29-30), 21679-21741. https://doi.org/10.1007/s11042-020-08862-1

Chan, K. Y., Abu-Salih, B., Qaddoura, R., Al-Zoubi, A. M., Palade, V., Pham, D.-S., Ser, J. D., & Muhammad, K. (2023). Deep neural networks in the cloud: Review, applications, challenges and research directions. *Neurocomputing*, *545*, 126327. https://doi.org/10.1016/j.neucom.2023.126327

Eminaga, O., Abbas, M., Shen, J., Laurie, M., Brooks, J. D., Liao, J. C., & Rubin, D. L. (2023). PlexusNet: A neural network architectural concept for medical image classification. *Computers in Biology and Medicine*, *154*, 106594. https://doi.org/10.1016/j.compbiomed.2023.106594

Houssein, E. H., Oliva, D., Samee, N. A., Mahmoud, N. F., & Emam, M. M. (2023). Liver Cancer Algorithm: A novel bio-inspired optimizer. *Computers in Biology and Medicine*, *165*, 107389. https://doi.org/10.1016/j.compbiomed.2023.107389

Jimenez-Mesa, C., Arco, J. E., Martinez-Murcia, F. J., Suckling, J., Ramirez, J., & Gorriz, J. M. (2023). Applications of machine learning and deep learning in SPECT and PET imaging: General overview, challenges and future prospects.

Pharmacological Research, *197*, 106984. https://doi.org/10.1016/j.phrs.2023.106984

Kaur, A., Singh, Y., Neeru, N., Kaur, L., & Singh, A. (2022). A Survey on Deep Learning Approaches to Medical Images and a Systematic Look up into Real-Time Object Detection. *Archives of Computational Methods in Engineering*, *29*(4), 2071-2111. https://doi.org/10.1007/s11831-021-09649-9

Liu, Z., Tong, L., Chen, L., Jiang, Z., Zhou, F., Zhang, Q., Zhang, X., Jin, Y., & Zhou, H. (2023). Deep learning based brain tumor segmentation: A survey. *Complex & Intelligent Systems*, *9*(1), 1001-1026. https://doi.org/10.1007/s40747-022-00815-5

Lundervold, A. S., & Lundervold, A. (2019). An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift Für Medizinische Physik*, *29*(2), 102-127. https://doi.org/10.1016/j.zemedi.2018.11.002

Mahmood, T., Rehman, A., Saba, T., Nadeem, L., & Bahaj, S. A. O. (2023). Recent Advancements and Future Prospects in Active Deep Learning for Medical Image Segmentation and Classification. *IEEE Access*, *11*, 113623-113652. https://doi.org/10.1109/ACCESS.2023.3313977

Martinez-Murcia, F. J., Ortiz, A., Ramírez, J., Górriz, J. M., & Cruz, R. (2021). Deep residual transfer learning for automatic diagnosis and grading of diabetic retinopathy. *Neurocomputing*, *452*, 424-434. https://doi.org/10.1016/j.neucom.2020.04.148

Metlek, S., & Çetıner, H. (2023). ResUNet+: A New Convolutional and Attention Block-Based Approach for Brain

Tumor Segmentation. *IEEE Access*, *11*, 69884-69902. https://doi.org/10.1109/ACCESS.2023.3294179

Ming, Y., Dong, X., Zhao, J., Chen, Z., Wang, H., & Wu, N. (2022). Deep learning-based multimodal image analysis for cervical cancer detection. *Methods*, *205*, 46-52. https://doi.org/10.1016/j.ymeth.2022.05.004

Mohd Adam, M. A., Ibrahim, A. I., Abidin, Z. Z., & Mohd Zaki, H. F. (2020). Deep Learning-Based Water Segmentation for Autonomous Surface Vessel. *IOP Conference Series: Earth and Environmental Science*, *540*(1), 012055. https://doi.org/10.1088/1755-1315/540/1/012055

Nam, D., & Pak, A. (2024). COMPARATIVE ANALYSIS OF U-NET, U-NET++, TRANSUNET AND SWIN-UNET FOR LUNG X-RAY SEGMENTATION. *Herald of the Kazakh-British Technical University*, *21*(2), 42-53. https://doi.org/10.55452/1998-6688-2024-21-2-42-53

Noor, M. H. M., & Ige, A. O. (2024). *A Survey on State-of-the-art Deep Learning Applications and Challenges* (No. arXiv:2403.17561). arXiv. https://doi.org/10.48550/arXiv.2403.17561

Ostmeier, S., Axelrod, B., Isensee, F., Bertels, J., Mlynash, M., Christensen, S., Lansberg, M. G., Albers, G. W., Sheth, R., Verhaaren, B. F. J., Mahammedi, A., Li, L.-J., Zaharchuk, G., & Heit, J. J. (2023). USE-Evaluator: Performance metrics for medical image segmentation models supervised by uncertain, small or empty reference annotations in neuroimaging. *Medical Image Analysis*, *90*, 102927. https://doi.org/10.1016/j.media.2023.102927

Peng, J., & Wang, Y. (2021). Medical Image Segmentation With Limited Supervision: A Review of Deep Network Models. *IEEE Access*, *9*, 36827-36851. https://doi.org/10.1109/ACCESS.2021.3062380

Pérez Cáceres, M., Gauvin, A., Dumais, F., & Iorio-Morin, C. (2024). A Practical Roadmap to Implementing Deep Learning Segmentation in the Clinical Neuroimaging Research Workflow. *World Neurosurgery*, *189*, 193-200. https://doi.org/10.1016/j.wneu.2024.06.026

Rahmani, M., Kazemi, A., Jalili Aziz, M., Najafzadeh, E., Farnia, P., & Ahmadian, A. (2024). A Systematic Review on Medical Image Segmentation using Deep Learning. *Scientia Iranica*, *0*(0), 0-0. https://doi.org/10.24200/sci.2024.61686.7441

Rodrigues, R., Lévêque, L., Gutiérrez, J., Jebbari, H., Outtas, M., Zhang, L., Chetouani, A., Al-Juboori, S., Martini, M. G., & Pinheiro, A. M. G. (2024). Objective quality assessment of medical images and videos: Review and challenges. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-024-20292-x

Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2024). A Comprehensive Overview and Comparative Analysis on Deep Learning Models. *Journal on Artificial Intelligence*, *6*(1), 301-360. https://doi.org/10.32604/jai.2024.054314

Siddique, N., Paheding, S., Elkin, C. P., & Devabhaktuni, V. (2021). U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications. *IEEE Access*, *9*, 82031-82057. https://doi.org/10.1109/ACCESS.2021.3086020

Soomro, T. A., Zheng, L., Afifi, A. J., Ali, A., Soomro, S., Yin, M., & Gao, J. (2023). Image Segmentation for MR Brain Tumor Detection Using Machine Learning: A Review. *IEEE Reviews in Biomedical Engineering*, *16*, 70-90. https://doi.org/10.1109/RBME.2022.3185292

Toğaçar, M. (2021). Detection of segmented uterine cancer images by Hotspot Detection method using deep learning models, Pigeon-Inspired Optimization, types-based dominant activation selection approaches. *Computers in Biology and Medicine*, *136*, 104659. https://doi.org/10.1016/j.compbiomed.2021.104659

Wang, X., Jing, S., Dai, H., & Shi, A. (2023). High-resolution remote sensing images semantic segmentation using improved UNet and SegNet. *Computers and Electrical Engineering*, *108*, 108734. https://doi.org/10.1016/j.compeleceng.2023.108734

Wang, Z., Zheng, J.-Q., Zhang, Y., Cui, G., & Li, L. (2024). *Mamba-UNet: UNet-Like Pure Visual Mamba for Medical Image Segmentation* (No. arXiv:2402.05079). arXiv. https://doi.org/10.48550/arXiv.2402.05079

Zeineldin, R. A., Karar, M. E., Coburger, J., Wirtz, C. R., & Burgert, O. (2020). DeepSeg: Deep neural network framework for automatic brain tumor segmentation using magnetic resonance FLAIR images. *International Journal of Computer Assisted Radiology and Surgery*, *15*(6), 909-920. https://doi.org/10.1007/s11548-020-02186-z

Zhou, Q., Wang, J., Yu, X., Wang, S., & Zhang, Y. (2023). A Survey of Deep Learning for Alzheimer's Disease. *Machine*

Learning and Knowledge Extraction, *5*(2), 611-668. https://doi.org/10.3390/make5020035

Zhou, S. K., Greenspan, H., Davatzikos, C., Duncan, J. S., Van Ginneken, B., Madabhushi, A., Prince, J. L., Rueckert, D., & Summers, R. M. (2021). A Review of Deep Learning in Medical Imaging: Imaging Traits, Technology Trends, Case Studies With Progress Highlights, and Future Promises. *Proceedings of the IEEE*, *109*(5), 820-838. https://doi.org/10.1109/JPROC.2021.3054390

Zhou, T., Porikli, F., Crandall, D. J., Van Gool, L., & Wang, W. (2023). A Survey on Deep Learning Technique for Video Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(6), 7099-7122. https://doi.org/10.1109/TPAMI.2022.3225573

BÖLÜM VIII

Etkili İlaç Yönetimi için Yenilikçi Bir Mobil Uygulama

Samet ÖZER[1]
Büşra TAKGİL[2]

Giriş

İlaç yönetimi, hastalıkların tedavisinde ilaçların güvenli ve etkili kullanımını içeren sağlık hizmetlerinin önemli bir basamağıdır. Bu süreç, tedavideki riskleri en aza indirirken optimum etki sağlayabilmesi için reçete yazma, dağıtma, uygulama, izleme ve ilaçları inceleme gibi çok çeşitli faaliyetleri kapsar. Modern farmakoterapinin artan karmaşıklığı ve kronik hastalıkların artan yaygınlığı ile etkili ilaç yönetimi, ilaç hatalarını azaltmada, istenmeyen ilaç reaksiyonlarını önlemede ve hasta uyumunu teşvik etmede esastır. Uygun ilaç yönetimi, hem güvenliği hem de yaşam kalitesini artıran tedavileri ve bireysel ihtiyaçlara göre uyarlamak

[1] Düzce Üniversitesi,Bilgisayar Mühendisliği Bölümü
[2] Dr. Öğr. Üyesi, Düzce Üniversitesi,Bilgisayar Mühendisliği Bölümü, 0000-0002-7927-0083:

için sağlık hizmeti sağlayıcıları, hastalar ve bakıcılar arasındaki iş birliğini içerir. Literatür incelendiğinde ilaç ve sağlık ile ilgili geliştirilen uygulamalarda ilaç hatırlatma ve sağlık takibi üzerine durulduğu gözlenmektedir. Geliştirilen bazı uygulamaların kapsamı sadece ilaç hatırlatma işlevleri üzerinde sınırlıdır (Patil vd., 2021).İlaç hatırlatma özelliğinin yanında adım takibi, kalori takibi, su takibi gibi sağlık takibi işlevleri ve nöbetçi eczane bulma gibi işlevler içeren uygulamalar da bulunmaktadır (Ameta vd., 2015: 39-48), (Güler ve Can, 2022: 7-12), (Saghaeiannejad-Isfahani vd., 2017: 108-11). Uygulamalar, tıbbi alanın tüm ortamlarında büyük bir potansiyele sahip olan ve sağlık hizmetlerinin kapsamını ve kalitesini iyileştirmek için umut verici fırsatlar sunan yeni bilgi teknolojilerinin araçları haline geliyor. Çoklu sensörlere sahip akıllı telefonlar bir başka ileri adımı oluşturuyor ve hem profesyoneller hem de hastalar tarafından kullanılmak üzere uygulamaların geliştirilmesine sebep olmaktadır (Posadilla vd., 2017: 227-236). Mobil uygulamalar, tıbbi yönetimle ilgili zorluklara yönelik çözümlerden birini temsil ediyor (Bailey vd., 2014: 542-546), çünkü 2018'de bile ABD'li yetişkinlerin %95'i bir cep telefonuna sahipti (Statista, 2024), (Pew Research Center, 2019). Dahası, insanlar günün büyük bir bölümünde cep telefonlarını yanlarında taşıyor, bu nedenle ilaç yönetimi uygulamaları hareket halindeyken ve gereken zamanda yardım sağlayabilmektedir (Brzan vd., 2016: 210-218). Mobil uygulamalar, hastalara kişiselleştirilmiş anında destek sağlama ve tedaviye uyumu ve uzun vadeli refaha katılımı iyileştirme potansiyeline sahiptir. Ancak, uygun bir ilaç yönetimi uygulaması aramak zordur, çünkü pazar yeri kullanıcının filtrelemesini zorlaştıran yüzlerce çözüm sunar. Güvenilir bir uygulama bulmak için, kullanıcının arama sırasında içerik kaynağı veya yazar bağlılığı hakkında kolayca bilgi sahibi olması faydalı olacaktır (Tabi vd., 2019). Yaşlılar ilaç kullanıcılarının büyük bir

çoğunluğunu oluşturmasına rağmen, mobil ilaç yönetimi uygulamaları genellikle daha genç popülasyonlar için tasarlanır. Sonuç olarak, işitme veya görme kaybı gibi yaşa bağlı fiziksel değişiklikler ve sezgisel olmayan tasarım özelliklerinin kullanımı, uygulamaların yaşlı kullanıcılar için yararlılığını sınırlar. İlaç kullanımını ve uyumu iyileştirmek için mobil müdahaleler geliştirirken, tasarımcılar, programcılar ve geliştiriciler yaşlı yetişkinleri potansiyel olarak yüksek etkili son kullanıcılar olarak değerlendirmeli ve bu popülasyonu tasarım sürecine dahil etmelidir. Yaşlılar için, hatırlatıcılar gibi standart özellikler çoğu mobil cihaza pek uygun olmayabilirken, yan etkiler veya ilaç etkileşimleri hakkında yüksek kaliteli bilgi sağlayan uygulamalar daha arzu edilir olabilir (Grindrod vd., 2014).

İlaç ve sağlık ile ilgili geliştirilen önceki uygulamalarda kullanıcının sağlık bilgilerine göre ilaçlar hakkında kişiselleştirilmiş bilgiler verilmemektedir. Bu yüzden kullanıcıyı kullandığı ilaçlar hakkında bilgilendirmekte geri kalmaktadırlar. Geliştirilen mobil uygulamadaki ilaç hatırlatıcı, ilaç kullanım geçmişi ve etkin madde analizi özellikleri önceki geliştirilen ilaç ve sağlık ile ilgili mobil uygulamaların özellikleriyle benzerlik göstermektedir. Günümüzde sağlık problemlerinin artmasıyla beraber ilaç kullanımı giderek artmaktadır. İlaç kullanımının artmasıyla birlikte ilaç kullanımında karmaşıklıklar meydana gelmektedir ve daha dikkatsiz ilaç kullanılmaktadır. İnsanlar genelde kullandıkları ilaçların prospektüslerini incelemeden ilaç kullanmaktadırlar. İlaç ve sağlık alanında geliştirilen diğer uygulamalarda kullanıcının sağlık bilgilerine dayalı ilaçlarla ilgili kişisel bilgilendirme yapılmamaktır. Bu yüzden diğer uygulamalar kullanıcıyı kullandığı ilaçlar hakkında bilgilendirmekte geri kalmaktadırlar. Geliştirilen uygulama kullanıcıya özel ilaç bilgileri özelliği bulunmaktadır. Bu özellik ile

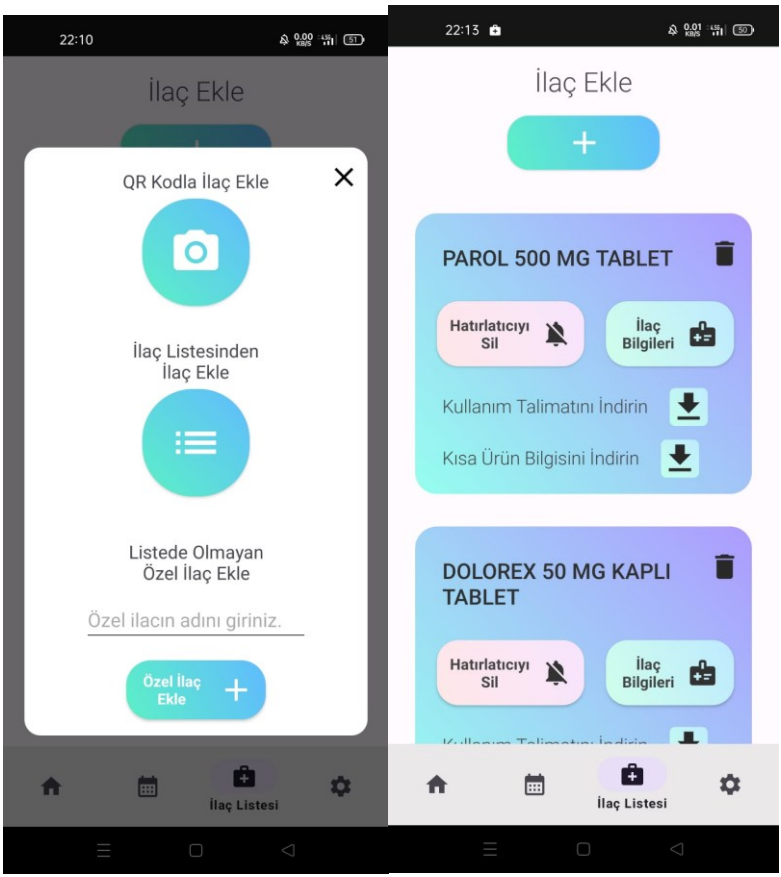
kullanıcıların sağlık bilgileri alınarak kullandıkları ilaçların prospektüs metinlerinden kendilerini ilgilendiren kısımlar çekilerek kullanıcıya sunulacaktır. Bu sayede kullanıcının kullandığı ilaçlar hakkında kısa yoldan bilgi sahibi olması sağlanacaktır Geliştirilen uygulamanın diğer uygulamalardan farkı sağlık bilgileri ile ilaç prospektüslerden kullanıcıyı ilgilendiren kısımlar çekerek kullanıcıya sunmasıdır. Bu sayede kullanıcı kullandığı ilaçlar hakkında kısa yoldan bilgi sahibi olmaktadır. Geliştirilen uygulama dahilindeki tüm özellikler ile kullanıcının ilaç yönetimini daha etkili bir şekilde yönetebilmesi hedeflenmektedir. Geliştirilen mobil uygulama, ilaç kullanan insanların ilaç yönetimini etkili bir şekilde gerçekleştirmesini sağlayacak bir araç olacaktır. Ayrıca uygulamada yer alan etkin madde çakışma analizi, ilaç hatırlatıcı ve ilaç kullanım geçmişi özellikleri ile ilaç yönetiminin daha etkili olması sağlanacaktır. Etkin madde çakışma analizi özelliği, kullanıcının uygulamaya eklediği ilaçların etkin maddeleri ile prospektüs metinlerini karşılaştırarak bir çakışma durumu tespit edildiğinde kullanıcıyı bilgilendiren bir özelliktir bu sayede kullanıcı ilaçların kullanımında ters bir durum olduğunda bilgi sahibi olmuş olur. İlaç hatırlatıcı özelliği ile kullanıcı kullandığı ilaçlar için hatırlatıcı kurabilir ve ilaç kullanımını düzenli şekilde gerçekleştirebilir. İlaç kullanım geçmişi özelliği ile kullanıcı hatırlatıcı kurulmuş ilaçların önceki kullanım durumlarını takvim üzerinden görüntüleyebilir ve ilaç kullanımını hakkında bilgi sahibi olmaktadır.

Yöntem

Geliştirilen mobil uygulama Android Studio kullanılarak Kotlin dili ile geliştirilmiştir. Uygulama Android 8.0 ve üstü sürümü bulunan cihazlarda çalışmak üzere geliştirilmiştir. Uygulama içinde 4 farklı sekmede kullanıcıya hizmet sunulmaktadır. Ana sayfa sekmesinde kullanıcının hatırlatıcı kurduğu ilaçların 14 gün

içerisindeki hatırlatıcı bilgileri yer almaktadır. Takvim sekmesinde kullanıcının geçmişte kullandığı ilaçların bilgileri yer almaktadır. İlaç listesi sekmesinde kullanıcının uygulamaya eklediği ve kullandığı ilaçlar listelenmektedir. Bu sekme içerisinde ilaçlar için hatırlatıcı kurma ve kullanıcıya özel ilaç bilgisi özellikleri bulunmaktadır. Ayarlar sekmesinde ise kullanıcı çeşitli sağlık bilgilerini düzenleyebilmektedir.

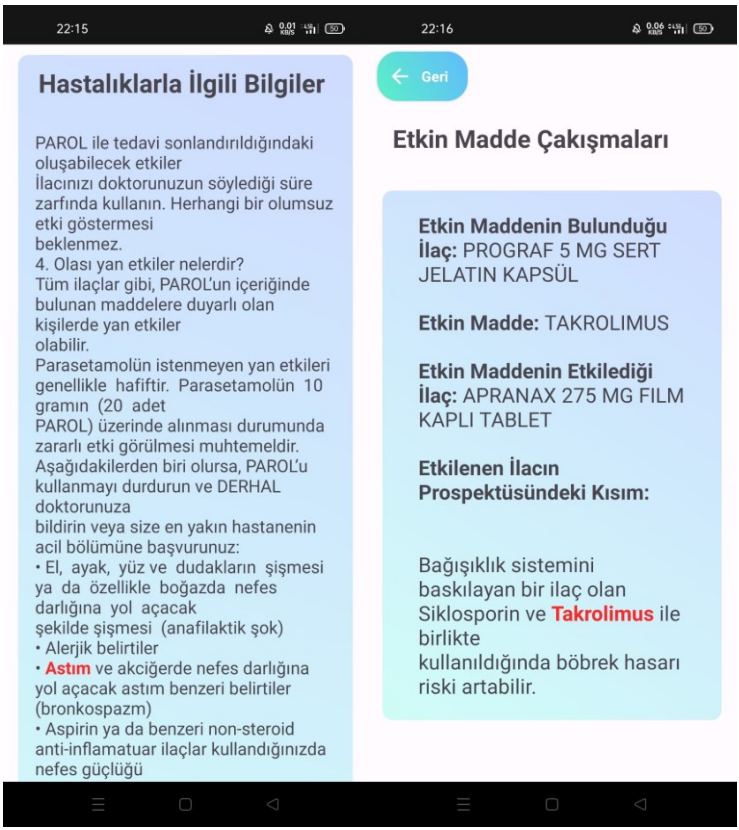
Bu çalışmada kullanılan ilaç verileri, Türkiye İlaç ve Tıbbi Cihaz Kurumu'nun resmi sitesinde herkese açık olarak yayımlanan ilaç verilerinden elde edilmiştir (TİTCK, 2024). Çalışma için resmi site üzerinde yayımlanan 2 veri setinden faydalanılmıştır. İlk veri setinde prospektüs dosyalarının linkleri bulunmaktadır. Bu veri seti Python kullanılarak web scraping yöntemi ile site üzerindeki tablodan elde edilmiştir. İkinci veri setinde ilaçların barkod numaraları bulunmaktadır. Bu veri seti site üzerinden Excel formatında bir dosya olarak elde edilmiştir. İki veri setindeki bilgiler Python ile birleştirilerek CSV formatında bir veri seti oluşturulmuştur. Uygulama içerisinde bulunan QR kod ile ilaç ekleme ve ilaç listesinden ilaç ekleme yöntemleri hazırlanan veri setinden bilgileri çekmektedir. Uygulama içindeki verilerin tamamı Room Database kullanılarak yerel veri tabanında tutulmaktadır. Kullanıcı bilgileri, kullanıcının kullandığı ilaçlar, hatırlatıcı bilgileri, ilaç kullanım geçmişi yerel veri tabanında saklanmaktadır. İlaçların prospektüs metinleri yerel veri tabanında tutulmaktadır. Kullanıcıya özel ilaç bilgileri ve ilaç etkin madde analizi işlevleri için "Regular Expression" yöntemi kullanılarak bu metinlerden cümleler çekilmektedir.



Şekil 1. a) İlaç ekleme penceresi b) İlaç listesi sekmesi

Şekil 1'de ilaç listesine ilaç yöntemleri olan Qr kod ile ilaç ekleme, ilaç listesinden ilaç ekleme ve özel ilaç ekleme gösterilmiştir. Bu yöntemlerden biri seçilip ilaç eklendiğinde Room Database üzerinden ilaç tablosuna ilaç verileri eklenmektedir ve bu sonrasında bu veriler ilaç listesi sayfasında listelenmektedir. Kullanım talimatı indirildiğinde prospektüs metni PDF içerisinden alarak ilaç verilerinde saklanır. İlaç için hatırlatıcı kurulup kurulmadı bilgisi, kullanım talimatının ve kısa ürün bilgisinin

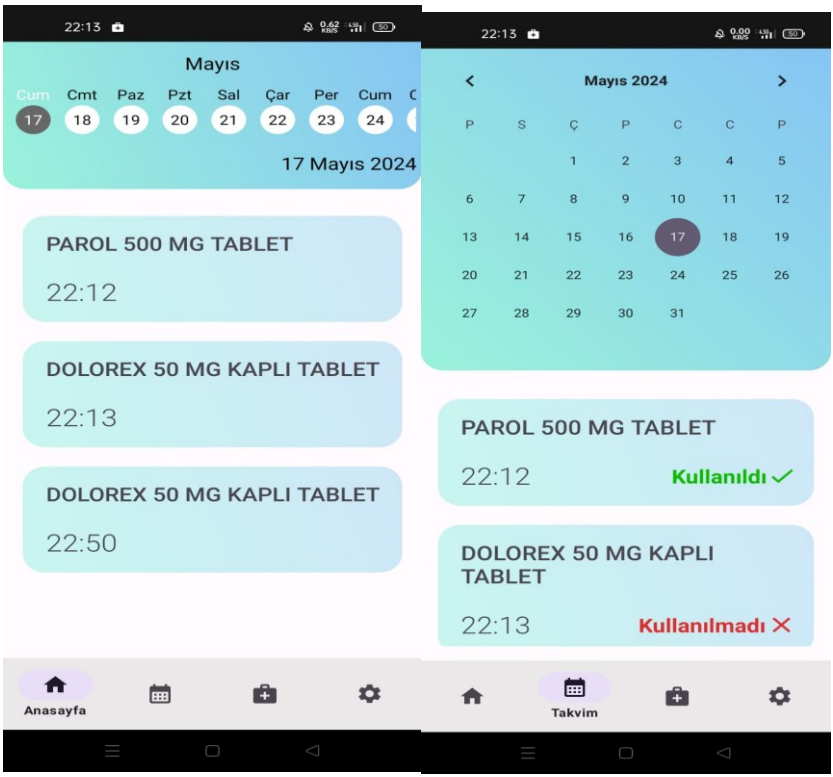
indirilip indirilmediği bilgisi tablodaki ilaç verisi içinde boolean değer olarak tutulmaktadır.



Şekil 2. a) Kullanıcıya özel ilaç bilgileri b) Etkin madde çakışması

Şekil 2'de kullanıcıya özel ilaç bilgileri ve etkin madde çakışmaları sayfaları gösterilmiştir. Kullanıcı ilaç bilgileri butonuna bastığında Room database üzerinden ilaç tablosundaki ilgili ilacın prospektüs metni alınır ve regular expression yöntemi ile kullanıcının sağlık bilgileri metin içerisinden alınır. Hastalık kelimelerinin geçtiği paragraflar alınarak kullanıcıya gösterilir. İlaç listesi sayfasına basıldığında Room Database üzerinden ilaçların

prospektüs metinleri ve ilaç etkin maddeleri çekilerek metinler içerisinde etkin madde kelimeleri taranır. Eşleşme bulunduğunda İlaç listesi sekmesinde bir uyarı belirir. Bu uyarıya basıldığında regular expression yöntemi ile prospektüs metninde etkin maddenin geçtiği paragraf alınarak kullanıcıya gösterilir.



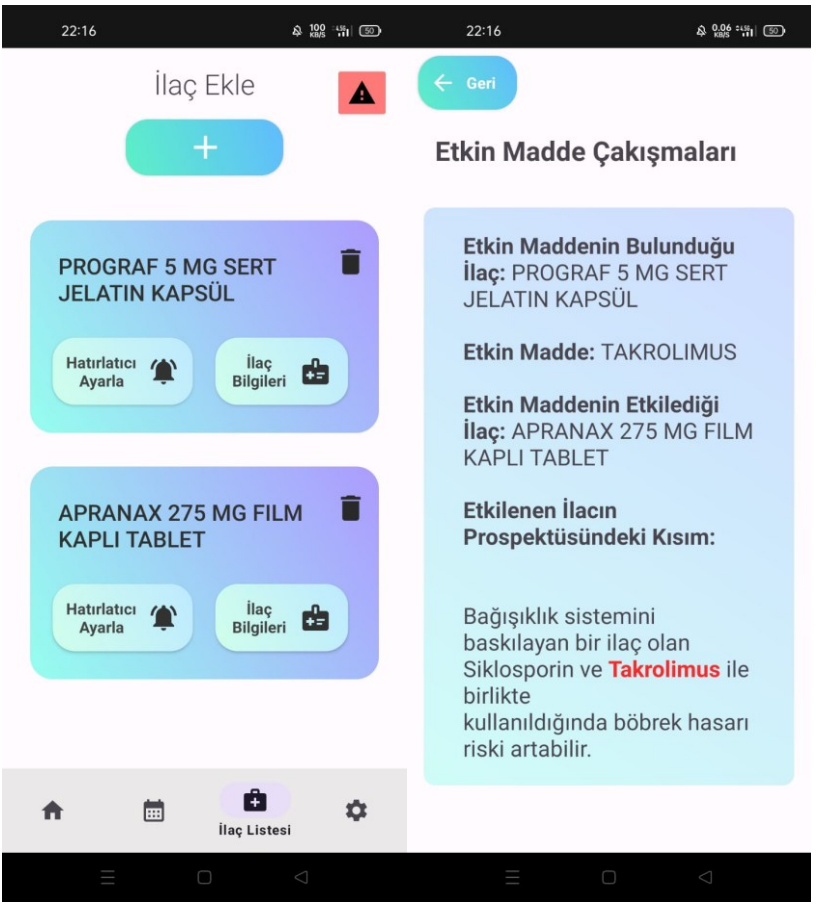
Şekil 3. a) Ana sayfa sekmesi b) Takvim sekmesi

Şekil 3'te Ana sayfa ve Takvim sekmelerinin görüntüsü verilmiştir. Bir Hatırlatıcı kurulduğunda bilgiler Room Database üzerinde alarm tablosu üzerinde tutulur. İlaçların kullanım bilgileri takvim tablosu üzerinde tutulmaktadır. Anasayfa sekmesindeki hatırlatıcı bilgileri alarm tablosundaki bilgilere göre göre listelenir.

Takvim sekmesinde kullanım bilgileri takvim tablosundaki bilgilere göre listelenir.

Bulgular ve Tartışma

Geliştirilen uygulama hedeflenen özellikleri başarıyla yerine getirmektedir. Uygulama, kullanıcılara sağlık bilgilerine göre ilaç bilgisi sunmakta ve bu sayede kullanıcılar kullandıkları ilaçlar hakkında bilinç sahibi olmaktadırlar. Bu özelliğin regular expression yöntemi ile sağlanması kullanıcıya sağlanan bilgiler için yeterli olmaktadır. İlaç hatırlatma özelliği ile kullanıcıların ilaçlarını düzenli şekilde kullanabilmeleri sağlanmıştır. İlaç kullanım geçmişi özelliği ile kullanıcıların ilaç kullanımlarını takip edebilmelerini sağlanmaktadır. Etkin madde çakışma analizi özelliği ile kullanıcıların kullandığı ilaçlar arasında tehlikeli etkileşim durumları olduğunda bilgilendirme yapılmaktadır. İlaç listesine birden fazla ilaç eklendiğinde ilaçların etkin maddeleri ve prospektüsleri eşleştirilerek etkin madde analizi yapılmaktadır. Room Database üzerinden kullanıcının eklediği ilaçların etkin madde bilgileri ve prospektüs metinleri alınarak etkin maddelerin prospektüs metinlerinin içinde bulunma durumu kontrol edilmektedir. Eğer çakışma durumu bulunursa İlaç Listesi sekmesinin sağ üstünde bir uyarı belirir ve bu uyarıya basıldığında etkin madde çakışma durumlarının listelendiği bir sayfa açılır. Bu sayfa içerisinde hangi ilaçlar arasında çakışma olduğu ve bu durumun prospektüs metni listelenir. Bir çakışma olduğunda beliren uyarı simgesi ve etkin madde çakışmalarının listelendiği sayfadaki örnek bir çakışma durumu Şekil 4’te gösterilmiştir.



Şekil 4. a) Çakışma durumunda uyarı işareti. b) Etkin madde çakışmaları sayfası

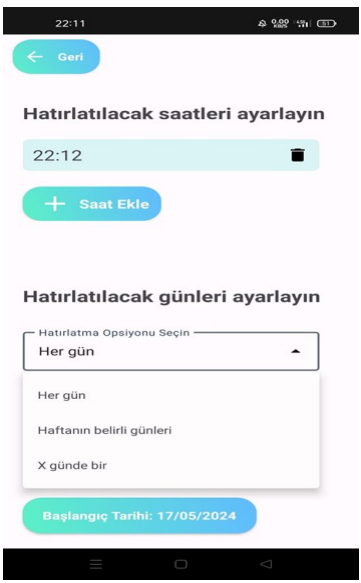
Geliştirilen mobil uygulama içerisindeki özellikler için kullanıcı dostu arayüz tasarımları tercih edilerek kullanıcı için kolay bir kullanım deneyimi amaçlanmıştır. Kullanıcının ilaç ekleyebilmesi için QR kod ile ilaç ekleme, ilaç listesinden ilaç ekleme ve özel ilaç ekleme opsiyonları sağlanmıştır. İlaç ve sağlık alanında geliştirilen diğer uygulamalarda olmayan sağlık bilgilerine

göre prospektüslerden bilgi çekilip kullanıcıya sunulması özelliği uygulamada etkili bir şekilde çalışmaktadır. Bu özellik sayesinde kullanıcılar kullandıkları ilaçlar hakkında kolay yoldan bilgi sahibi olurlar. İlaç etkin madde analizi, ilaç hatırlatıcı ve ilaç kullanım geçmişi özellikleri uygulamaya dahil edilmiştir ve başarılı şekilde çalışmaktadırlar. Uygulama için belirlenen Anasayfa, Takvim, İlaç listesi ve Ayarlar sekmeleri uygulamadaki tüm işlevleri içerecek şekilde tasarlanmış ve kullanıcı dostu arayüz ile kullanıcıya hizmet sağlanmıştır.

Bir ilaç listeye eklendikten sonra "Hatırlatıcı Ayarla" butonu ile ilaç için hatırlatıcı ayarlanabilmektedir. Hatırlatıcı ayarlanan gün ve saatlerde kullanıcıya bildirim gönderilmektedir. Butona tıklandıktan sonra açılan sayfada ilk olarak Saat Ekle butonu ile gün içerisinde hangi saatlerde hatırlatıcının bildirim göndereceği seçilir. Hatırlatıcı bir güç içinde bir veya birden fazla saate kurulabilmektedir. Sonrasında hatırlatma opsiyonu seçme bölümü açılmaktadır. Bu bölümde Her gün, Haftanın belirli günleri ve X günde bir bildirim gönderilecek şekilde 3 adet opsiyon bulunmaktadır. Her gün opsiyonu seçildiğinde belirlenen saatlerde her gün bildirim gönderilmektedir. Haftanın belirli günleri opsiyonu seçildiğinde haftanın günlerinin olduğu bir liste gelmektedir ve kullanıcının listeden seçtiği günlerde belirlenen saatlerde bildirim gönderilmektedir. X günde bir opsiyonu seçildiğinde kullanıcıdan bir sayı seçmesi beklenir ve kullanıcının seçtiği gün aralığına göre bildirim gönderilir. Hatırlatma sıklığı belirlendikten sonra bildirimin başlangıç tarihinin seçildiği bölüm açılır. Bu bölümde seçilen tarihten itibaren hatırlatıcı çalışmaya başlar. Tüm gerekli bilgiler alındıktan sonra onayla butonuna basıldığında ilaç için hatırlatıcı

ayarlama işlemi tamamlanır. Hatırlatıcı ayarlandıktan sonra Hatırlatıcı Ayarla butonu değişerek Hatırlatıcı Sil butonu olmaktadır. Bu butona basıldığında ayarlanan hatırlatıcı silinmektedir. Bir ilaç için gönderilmiş bildirim Şekil 5'te gösterilmiştir. Hatırlatıcı Ayarla sayfası Şekil 6'da gösterilmiştir.

Şekil 5. İlaç bildirimi.



Şekil 6. Hatırlatıcı Ayarla sayfası

Hatırlatıcı kurulduktan sonra hatırlatıcı ayarları ve sonrasına hatırlatıcı bildirimi gönderildiğinde kullanım bilgisi Room Database ile yerel veri tabanında saklanmaktadır. Veri tabanındaki bu bilgiler ile Anasayfa ve Takvim sekmelerindeki listelere bilgi gönderilmektedir. Hatırlatıcı algoritması olarak batarya tasarrufu sağlayacak bir sistem kullanılmıştır. Hatırlatıcı Ayarla sayfasından bir hatırlatıcı ayarlandığında sadece sonraki bildirim gönderilecek tarihe hatırlatıcı kurulur ve sonrasında bildirim gönderildiğinde bir sonraki bildirim gönderilecek tarihe hatırlatıcı kurulur. Bu sayede kullanıcının seçtiği opsiyonlara göre gönderilecek bildirimler için aynı anda birden fazla hatırlatıcı olması yerine aynı anda sadece bir hatırlatıcı olmaktadır. Bu sistem sayesinde telefonda batarya tasarrufu sağlanır.

Geliştirilen uygulamadaki etkin madde çakışma analizi, ilaç hatırlatıcı ve ilaç kullanım geçmişi özellikleri literatürdeki diğer ilaç ve sağlık ile ilgili geliştirilen uygulamalarla benzerlik göstermektedir. Kullanıcıya özel ilaç bilgileri özelliği ile literatürdeki diğer uygulamalardan farklı bir işlev ortaya koymaktadır. Bu özellik sayesinde kullanıcı sağlık bilgilerine göre ilaç hakkında kendini ilgilendiren bilgilere kolay yoldan erişim sağlar. Bu özellik doğal dil işleme yöntemleri ve yapay zeka kullanılarak geliştirilebilir durumdadır. Uygulamadaki ilaç veri seti geliştirilebilir durumdadır. Farklı kaynaklar kullanılarak daha gelişmiş bir ilaç veri seti elde edilebilir. Uygulamada kullanılan ilaç veri tabanı Türkiye İlaç ve Tıbbi Cihaz kurumunun resmi sitesinden alınmıştır. Bu veri tabanında kullanım talimatı ve kısa ürün bilgisi belgelerinde çeşitli eksiklikler yer almaktadır. Bu eksikliklerin tamamlanması uygulamanın performansını da arttıracaktır. QR kod ile ilaç ekleme opsiyonu için resmi siteden alınan 2 ilaç listesinde

eşleştirme yapılarak barkodlar ve ilaç isimleri eşleştirilerek yapılmaktadır. Eşleştirilen ilaçların sayısının artması yine uygulamanın işlevini arttıracaktır. Uygulamada yer alan ilaç veri tabanı geliştirilmeye, güncellemeye açıktır ve yapay zeka, diğer doğal dil işleme metotları kullanılarak yapılabilmektedir.

Sonuçlar

Bu çalışmada ilaç yönetimini kolaylaştıran bir mobil uygulama geliştirilmiştir. Geliştirilen uygulama kullanıcı dostu bir arayüze sahiptir ve özellikle yaşlı bireyler için bile kolay bir kullanım sağlanmıştır. Uygulamada kullanıcının kullandığı ilaçları ekleyebilmesi için planlanan QR kod ile ekleme özelliği sayesinde , ilaç listesinden ekleme ve özel ilaç ekleme işlemleri yapılabilmektedir. Sağlık bilgilerine göre prospektüsten ilaç bilgilerinin çekilip sunulması işlevi başarılı şekilde çalışmaktadır. Hatırlatıcı ayarlama işlevleri ve hatırlatma geçmişi uygulamanın işlevsel ve güçlü yönlerindendir. Etkin madde çakışma analizi sistemi kullanıcıyı doğru şekilde bilgilendirmektedir. Uygulama için belirlenen Anasayfa, Takvim, İlaç listesi ve Ayarlar sekmeleri uygulamadaki tüm işlevleri içerecek şekilde tasarlanmış ve kullanıcı dostu bir ara yüz ile hizmet sağlamaktadır. Uygulama içerisindeki tüm hizmetler kullanıcı kullanımına hazır durumdadır ve ilaç yönetimini daha etkili bir şekilde yönetebilmesi hedeflenmektedir. Geliştirilen mobil uygulama, ilaç kullanan insanların ilaç yönetimini etkili bir şekilde gerçekleştirmesini sağlayacak bir araç olacaktır. Ayrıca uygulamada yer alan etkin madde çakışma analizi, ilaç hatırlatıcı ve ilaç kullanım geçmişi özellikleri ile ilaç yönetiminin etkisini arttırmaktadır. Etkin madde çakışma analizi özelliği, kullanıcının uygulamaya eklediği ilaçların etkin maddeleri ile prospektüs metinlerini karşılaştırarak çakışma tespiti sayesinde kullanıcı önemli öngörülere sahip olabilmektedir. İlaç hatırlatıcı

özelliği sayesinde ise kullanıcı kullandığı ilaçlar için hatırlatıcı kurabilir ve ilaç kullanımını düzenli şekilde gerçekleştirebilir. İlaç kullanım geçmişi özelliği ile kullanıcı hatırlatıcı kurulmuş ilaçların önceki kullanım durumlarını takvim üzerinden görüntüleyebilir ve ilaç kullanımını hakkında bilgi sahibi olmaktadır. Tüm bu özellikler kullanıcıları kullandıkları ilaçlar hakkında daha bilinçli hale getirmektedir. Uygulamadaki diğer özellikler sayesinde ise ilaç kullanımı etkili şekilde yönetebileceklerdir. Uygulama Google Play Store'a yüklenerek kullanıcı kullanımına açık hale getirilebilecektir. Bu sayede kullanıcılar rahatlıkla mobil cihazlardan uygulamaya erişip ilaç deneyimlerini iyileştirebilirler.

Kaynaklar

Ameta, D., Mudaliar, K., & Patel, P. (2015). Medication reminder and healthcare – An Android application. *International Journal of Managing Public Sector Information and Communication Technologies, 6*(2), 39-48. https://doi.org/10.5121/ijmpict.2015.6204

Bailey, S. C., Belter, L. T., Pandit, A. U., Carpenter, D. M., Carlos, E., & Wolf, M. S. (2014). The availability, functionality, and quality of mobile applications supporting medication self-management. *Journal of the American Medical Informatics Association, 21*(3), 542-546.

Brzan, P. P., Rotman, E., Pajnkihar, M., & Klanjsek, P. (2016). Mobile applications for control and self-management of diabetes: A systematic review. *Journal of Medical Systems, 40*(9), 210-218.

Grindrod, K. A., Li, M., & Gates, A. (2014). Evaluating user perceptions of mobile medication management applications with older adults: A usability study. *JMIR mHealth and uHealth, 2*(1), e3048.

Güler, O., & Can, Ö. (2022). QuaVita: Sağlıklı yaşam mobil uygulaması. *Avrupa Bilim ve Teknoloji Dergisi, 44*, 7-12. https://doi.org/10.31590/ejosat.1216686

Patil, S. A., Bhanuse, M., Mali, S., & Swami, V. (2021). Review on mobile application for medicine reminder. *International Research Journal of Modernization in Engineering Technology and Science, 3*(2). Erişim: 23 Mayıs 2024, www.irjmets.com

Pew Research Center. (2019). Mobile fact sheet. Erişim adresi: http://www.pewinternet.org/fact-sheet/mobile/

Posadilla, D., Gómez-Marcos, V., & Hernández-Tejedor, A. (2017). Apps and intensive care medicine. *Medicina Intensiva, 41*(4), 227-236.

Saghaeiannejad-Isfahani, S., Ehteshami, A., Savari, E., & Samimi, A. (2017). Developing the medication reminder mobile application 'seeb'. *Acta Informatica Medica, 25*(2), 108-111. https://doi.org/10.5455/aim.2017.25.108-111

Statista. (2024). Number of mobile phone users worldwide from 2015 to 2020 (in billions). Erişim adresi: https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/

Tabi, K., Randhawa, A. S., Choi, F., Mithani, Z., Albers, F., Schnieder, M., Nikoo, M., Vigo, D., Jang, K., Demlova, R., & Krausz, M. (2019). Mobile apps for medication management: Review and analysis. *JMIR mHealth and uHealth, 7*(9), e13608.

Türkiye İlaç ve Tıbbi Cihaz Kurumu (TİTCK). (2024). Erişim: 23 Mayıs 2024, https://www.titck.gov.tr/kubkt

BÖLÜM IX

Makine Öğrenmesi: Yöntemler, Algoritmalar Ve Gerçek Dünya Uygulamaları

Şükran YAMAN ATCI[1]
Fatma KOSOVALI ÇAVUŞ[2]
Zeynep İNEL ÖZKİPER[3]
Sara NAGHIBZADEH[4]

1.Giriş

Makine öğrenmesi, veri analizi ve karar verme süreçlerinde devrim yaratarak son on yılın en etkili teknolojilerinden biri haline gelmiştir. Bu teknoloji, verilerden otomatik olarak öğrenme yeteneğine dayanarak, geniş bir yelpazede uygulama potansiyeline sahiptir. Modern makine öğrenmesi yaklaşımları, insan

[1] Dr. Öğr. Üyesi., Haliç Üniversitesi, Bilgisayar Programcılığı Bölümü, 34060, İstanbul/Türkiye, Orcid: 0000-0002-6600-4157, sukranatci@halic.edu.tr
[2] Dr. Öğr. Üyesi., Haliç Üniversitesi, Elektronik Teknolojileri Bölümü, 34060, İstanbul/Türkiye, Orcid: 0000-0003-2725-2876, fatmacavus@halic.edu.tr
[3] Öğr. Gör., Haliç Üniversitesi, Bilgisayar Programcılığı Bölümü, 34060, İstanbul/Türkiye Orcid: 0000-0003-3044-0317, zeynepinel@halic.edu.tr
[4] Öğr. Gör., Haliç Üniversitesi, Bilgisayar Programcılığı Bölümü, 34060, İstanbul/Türkiye, Orcid: 0009-0005-6959-1165, sarazadeh@halic.edu.tr

müdahalesini azaltarak ve algoritmik karmaşıklığı yöneterek, veri analizi, desen tanıma ve karar verme süreçlerini dönüştürmüştür. Bu çalışma makine öğrenmesinin kavramsal çerçevesini, ana yöntemlerini ve uygulama alanlarını derinlemesine tartışmaktadır. Ayrıca, sektörel uygulamaların yanı sıra akademik araştırmalarda da makine öğrenmesinin nasıl kritik bir rol oynadığına dikkat çekmektedir.

Çalışmada, makine öğrenmesinin tanımı ve tarihsel gelişimi ele alınmaktadır. Bu, algoritmanın temelini oluşturan teorik modellerin ve matematiksel prensiplerin yanı sıra, makine öğrenmesinin evrimini de kapsamaktadır. İlk uygulamalardan günümüzdeki derin öğrenme devrimine kadar olan süreç, bu alandaki yeniliklerin nasıl hızlı bir şekilde evrildiğini göstermektedir. Çalışmada makine öğrenmesi algoritmalarının sınıflandırılması yapılmakta ve bunların başlıca üç kategori altında toplandığı belirtilmektedir: gözetimli öğrenme, gözetimsiz öğrenme ve pekiştirmeli öğrenme. Her bir kategori, algoritmanın işleyişi, uygulandığı problem türleri ve tipik uygulama senaryoları ile detaylı bir şekilde açıklanmaktadır. Veri kalitesinden algoritma önyargılarına, hesaplama maliyetlerinden modelin genelleştirme yeteneğine kadar bir dizi faktör, makine öğrenmesi projelerinin başarısını etkileyebilmektedir. Bu faktörlerin her biri, algoritmanın performansını doğrudan etkileyen kritik öneme sahiptir.

Son olarak, makine öğrenmesinin geleceği üzerine düşünceler ve bu alandaki potansiyel gelişim yolları tartışılmaktadır. Özellikle yapay zekâ ve makine öğrenmesinin etik boyutları, bu teknolojilerin daha geniş kabul ve uygulama alanı bulmasındaki önemli faktörlerden biri olarak değerlendirilmektedir. Bu çalışma, makine öğrenmesinin teorik ve pratik yönlerini ele alarak, gelecekteki teknolojik gelişmelere ışık tutacak önemli bir potansiyel sunduğunu ortaya koymaktadır.

2.Yöntem

Makine öğrenmesi, kullanılan algoritmalar ve programlama dilleriyle, karmaşık problemlere çözüm sunan bir yaklaşım olarak öne çıkmaktadır. Algoritmaların performansını değerlendirmek için

genellikle hazır veri setleri tercih edilir. Bu veri setlerinin etkili bir şekilde kullanımı, belirli hazırlık süreçlerini gerektirir

Veri hazırlama süreci, veri temizleme, veri entegrasyonu, veri dönüşümü ve veri indirgeme aşamalarından oluşmaktadır. Veri temizleme aşamasında eksik veriler, tutarsız veriler ve aykırı değerler tespit edilerek uygun yöntemlerle düzeltilir. Veri entegrasyonu aşamasında farklı veri kaynaklarından elde edilen veriler birleştirilir ve tutarlı hale getirilir. Veri dönüşümü aşamasında veriler, normalizasyon ve ölçekleme teknikleri kullanılarak makine öğrenmesi algoritmalarına uygun hale getirilir. Veri indirgeme aşamasında ise gereksiz değişkenler ve özellikler çıkarılarak veri seti optimize edilir.

Makine öğrenmesi modelleri, seçilen algoritmalar ve programlama dilleri kullanılarak oluşturulur ve eğitilir. Her bir algoritma ve dil için algoritma seçimi, model eğitimi ve model değerlendirmesi adımları izlenir. Denetimli öğrenme için k-En Yakın Komşu, Yapay Sinir Ağları, Destek Vektör Makineleri gibi algoritmalar; denetimsiz öğrenme için K-Means ve PCA gibi algoritmalar kullanılabilir. Eğitim veri seti kullanılarak algoritmalar eğitilir ve hiperparametre optimizasyonu yapılabilir. Modellerin performansı, çapraz doğrulama teknikleri kullanılarak değerlendirilir.

Makine öğrenmesi modellerinin performansı, doğruluk, F1 skoru, hata oranı ve ROC AUC gibi metrikler kullanılarak değerlendirilebilir. Denetimsiz öğrenme algoritmalarının performansı ise kümeleme performansı (Silhouette Skoru, Davies-Bouldin İndeksi) ve boyut azaltma başarımı (Explained Variance Ratio) ile ölçülür.

Programlama dillerinin makine öğrenmesi üzerindeki etkileri, geliştirme kolaylığı, kütüphane desteği, performans ve topluluk desteği kriterleri doğrultusunda değerlendirilir. Geliştirme kolaylığı, kod yazma süresi, hata ayıklama ve hata yönetimi gibi unsurları kapsamaktadır. Kütüphane desteği, makine öğrenmesi kütüphanelerinin çeşitliliği ve işlevselliği ile değerlendirilir. Performans kriteri, eğitim süresi, bellek kullanımı ve işlemci yükü

ile ölçülür. Topluluk desteği ise geliştirici topluluğunun büyüklüğü ve aktifliği, dokümantasyon kalitesi ile değerlendirilebilir.

Makine öğrenmesi modellerinin gerçek dünya uygulamaları ve vaka çalışmaları, finans, sağlık, perakende ve akıllı şehirler gibi sektörlerde incelenmiştir. Bu inceleme, literatür taraması ve sektörel örnekler üzerinden gerçekleştirilmiştir.

Son olarak, makine öğrenmesinin gelecekteki potansiyel gelişim alanları ve araştırma yönleri hakkında öneriler sunulmuştur. Makine öğrenmesinin kullanıldığı alanlarla ilgili kapsamlı bilgiler sunulmuştur. Bu yöntem bölümü, çalışmanın kapsamlı ve sistematik bir şekilde gerçekleştirilmesini sağlayarak, makine öğrenmesi algoritmalarının ve programlama dillerinin etkinliği hakkında bilgi vermeyi amaçlamaktadır.

3.Bulgular

4.1. Makine Öğrenmesi

Öğrenme, Türk Dil Kurumu'na göre bilgi toplama, hatırlama ve beceriler edinme sürecidir (Türk Dil Kurumu, 2024). İnsanlar yaşamın ilk anlarından itibaren öğrenme kabiliyeti geliştirir ve bu yeteneklerini sürekli genişletirler. Yapay zekâ çalışmaları, insan zekasını makinelerle bütünleştirme fikri üzerine kurulmuştur. Bu bağlamda, makine öğrenmesi, makinelerin insanlar gibi öğrenebilmesi ve doğru kararlar verebilmesi arzusuyla doğmuştur. Başlangıçta bilgisayarlar belirli görevleri yerine getirmek üzere tasarlanmışken, zamanla veri toplama ve saklama yetenekleri gelişmiştir. Ancak veriler analiz edilip anlamlandırılmadıkça bilgiye dönüşmezler. "Elde edilen gözlemlerden toplanan veriler bilgisayar aracılığıyla işlenip, bu gözlemlerde bulunan örüntüler anlamlandırılabilir mi?" sorusu makine öğrenmesinin temel sorularından biri olmuştur (Emir Ş., 2013). Makine öğrenmesi, örnek verileri veya önceki deneyimleri kullanarak performansını en uygun ve verimli hale getiren bir bilgisayar programı olarak tanımlanır (Alpaydın, 2004).

Makine öğrenmesi yöntemleri, temelde denetimli ve denetimsiz öğrenme olarak iki ana kategoriye ayrılır. Eklenen yeni

yöntemler arasında yarı denetimli öğrenme, pekiştirmeli öğrenme, transdüktif öğrenme, çevrimiçi öğrenme ve aktif öğrenme bulunur.

Denetimli Öğrenme: Eğitim setindeki girdi ve çıktıların bir denetmen tarafından incelendiği bir öğrenme şeklidir (Öztemel, 2012). Bu süreçte, her bir örneğe ait girdiler ve bunlara karşılık gelen çıktılar sisteme tanıtılır. Denetimli öğrenmenin amacı, eğitim seti temelinde oluşturulan fonksiyon ile elde edilen çıktılar ile gerçek çıktılar arasındaki farkı minimize etmektir. Genellikle sınıflandırma ve regresyon temelli tahmin problemlerinde kullanılır.

Denetimsiz Öğrenme: Denetimsiz öğrenme, denetçi olmaksızın yalnızca girdi değişkenleri ile çalışan bir öğrenme metodudur. Bu yöntemde, sistem girdiler arasındaki ilişkileri kendi başına keşfetmekle yükümlüdür. Örneğin, bir havayolu şirketi müşteri memnuniyet anketlerini kullanarak çeşitli müşteri grupları oluşturabilir. Denetimsiz öğrenme, kümeleme ve boyut azaltma gibi alanlarda kullanılır.

Yarı Denetimli Öğrenme: Yarı denetimli öğrenme, girdi değişkenlerine sahip çok sayıda gözlem ile az sayıda hem girdi hem de çıktı değişkenlerine sahip gözlemlerin bir arada bulunduğu veri kümeleri için kullanılır. Bu yöntem, özellikle sınıflandırma ve regresyon gibi problemleri çözmek için kullanılır (Mohri vd., 2012).

Pekiştirmeli Öğrenme: Pekiştirmeli öğrenme, sistemlere verilen hedefleri deneme-yanılma yöntemiyle ulaşmayı öğreten bir yöntemdir ve dinamik programlama ile denetimli öğrenme tekniklerini bir arada kullanır (Kartal, 2015). Her eylem sonucunda sistem, aldığı ödül veya ceza ile bir sonraki adımda daha yüksek ödül elde etmeyi hedefler. Bu yöntem, robotların yönlendirilmesi ve gerçek zamanlı karar alma süreçlerinde kullanılır.

Makine öğrenmesi süreci, belirli bir problemin başarılı bir şekilde çözülmesi için izlenmesi gereken temel adımlara dayanır (Shearer, 2000):

Problemin Tanımlanması: Makine öğrenmesi projelerinde problemin net şekilde ifade edilmesi, tüm sürecin başarısını doğrudan etkileyen ilk ve en kritik adımdır.. Problem doğru

tanımlanmazsa, araştırmacılar hangi yönde ilerlemeleri gerektiği konusunda belirsizlik yaşar (Shearer, 2000).

Veri Analizi: İkinci adım, problem için uygun verilerin toplanmasıdır. Veriler, var olan veri ambarları veya veri tabanlarından elde edilebilir veya araştırmacının kendisi tarafından toplanabilir. Verilerin formatı, miktarı ve özellikleri incelenir (Shearer, 2000).

Verilerin Hazırlanması: Toplanan verilerin algoritmalar tarafından işlenebilir hale getirilmesi. Bu süreçte veri temizleme, entegrasyon, dönüşüm ve indirgeme işlemleri yapılır. Eksik veriler tamamlanır, veriler standardize edilir ve gereksiz değişkenler azaltılır.

Modelin Kurulması: Model kuruluken ulaşılabilecek en iyi sonuç için kullanılacak metodun seçilmesi deneme yanılma yöntemiyle belirlenir. Veri setindeki ilişkileri en doğru biçimde yansıtabilecek model ve algoritmalar tespit edilir (Hornick, Marcadé ve Venkayala, 2007).

Modelin Değerlendirilmesi: Elde edilen sonuçların kalitesinin değerlendirilmesi ve önceden belirlenen hedeflere ne kadar ulaşıldığının ölçülmesi (Demircioğlu ve Bilge, 2015). Farklı modellerin performansı karşılaştırılır ve genelleyebilirlik testleri yapılır.

Modelin Kullanılması: Seçilen model, problemin temel çözümü olarak kabul edilir ve benzer problemleri çözmek için kullanılabilir. Modelin yaygın kullanımı, başarı oranını artırır.

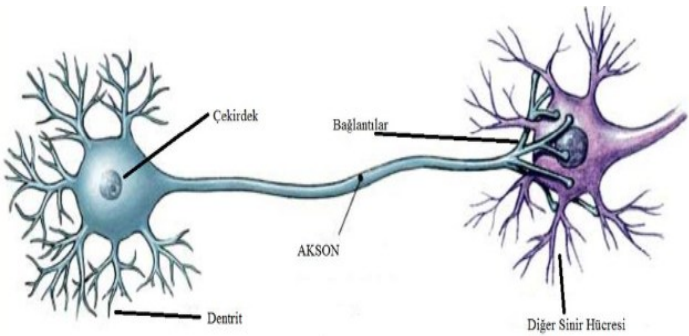
Bu adımlar, makine öğrenmesi sürecinde belirlenen problem hakkında derinlemesine bir anlayış geliştirmek ve veri setleri üzerinde etkili çalışmalar yapmak için kritiktir.

4.2. Makine Öğrenmesinde Kullanılan Algoritmalar

k-En Yakın Komşu (k-NN), 1951'de Fix ve Hodges'in geliştirdiği, benzerlik ve mesafe ölçütlerine dayalı basit bir makine öğrenme yöntemidir. Bu algoritma, yeni bir test örneğinin, var olan tüm örneklerle arasındaki mesafeleri hesaplar ve en yakın komşuları

belirler. Öklid Uzaklığı gibi çeşitli uzaklık ölçüm teknikleri kullanılır. Örneğin, k=3 seçildiğinde, test örneğine en yakın üç nokta incelenir ve en sık rastlanan sınıfa test örneği atanır.

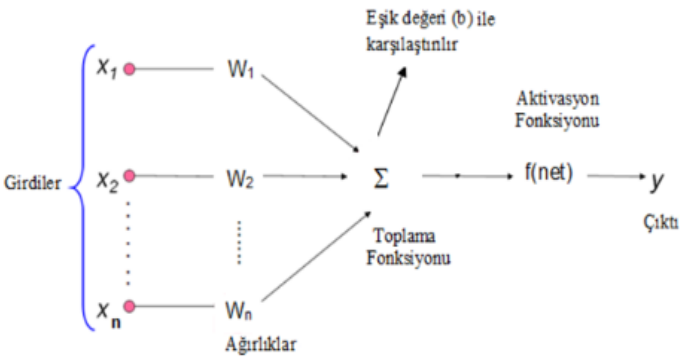
Yapay Sinir Ağları (YSA), insan beyninin sinirsel yapısını taklit eden matematiksel modellerdir. Biyolojik nöron hücresi Şekil 1'de sunulmuştur



Şekil 1 : *Biyolojik nöron hücresi*

(Mijwel, 2017).

Yapay sinir hücresinin işleyişi, Şekil 2'de detaylandırılmıştır.



Şekil 2 : *Yapay nöron hücresi*

(Öztürk ve Şahin 2018).

Bu yapı, girdilerin belirli ağırlıklarla işlemden geçirilmesini ve çıktıya dönüştürülmesini sağlar. Bu süreç, makine öğrenmesinin

temel yapı taşlarından biri olarak kabul edilir. YSA'lar, nöronların bağlantı biçimlerine göre ikiye ayrılır; ileri beslemeli ağlar ve geri beslemeli ağlar.

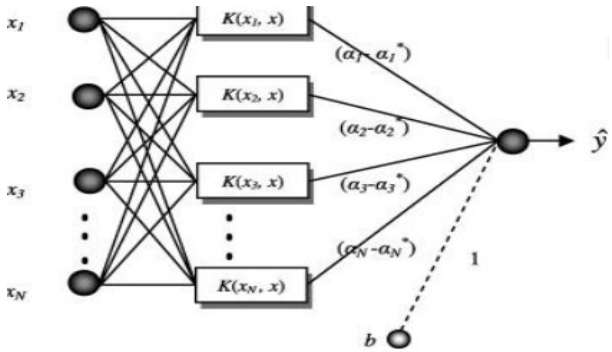
Naive Bayes Sınıflandırıcısı, olasılık temelli bir algoritmadır ve değişkenler arasında bağımsızlık varsayımına dayanır. Bu yöntem, hızlı ve basit bir şekilde sınıflandırma yapmak için kullanılır. (Mohssen ve diğ., 2016). Veri setindeki örnekler ve mevcut sınıf kümeleri göz önünde bulundurularak, Naive Bayes Sınıflandırıcısının amacı P(Ci|X) olasılığını maksimize etmektir (Yang, 2018). Değişkenler arası bağımsızlık varsayımı kullanılarak P(X|Ci) hesaplaması yapılır (Berrar, 2019).

Karar ağaçları, veri setlerini analiz ederek sınıflandıran bir öğrenmesi yöntemidir. Veriler kökten yapraklara doğru ilerler ve her düğüm belirli bir özelliğe göre verileri dallara ayırır.

Karar ağaçları, eğitim ve test aşamalarının hızlı gerçekleşmesi, sonuçların kolayca yorumlanabilir olması ve görsel olarak ifade edilebilmesi gibi avantajlara sahiptir (Özgür ve Erdem, 2012).

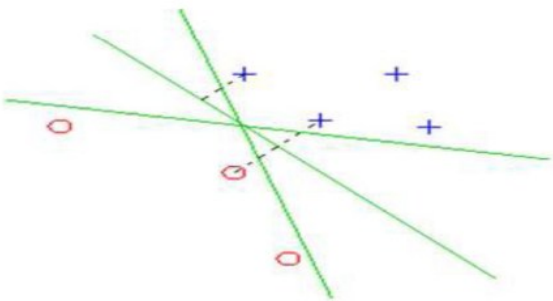
Random Forest, Leo Breiman tarafından geliştirilen ve karar ağaçlarının bir araya gelmesiyle oluşturulan bir makine öğrenmesi metodudur. Bu yöntemde, bootstrap yöntemiyle rastgele örnekler çıkarılır ve her düğümde kullanılacak az sayıda değişken seçilir (Akman, 2010). Random Forest, aşırı uyuma karşı dayanıklıdır ve büyük veri setlerinde etkili çalışır (Breiman ve Cutler, 2011).

Destek Vektör Makineleri (DVM), veri setlerini ayırmak için kullanılan etkili bir sınıflandırma metodudur. İki sınıflı ve çok sınıflı problemler için uygundur (Ayhan ve Erdoğmuş, 2014). Destek vektör makinelerinin temel yapısı Şekil 3’de verilmiştir.



Şekil 3 : *Destek Vektör Makinelerinin (DVM) temel yapısı. (Ceryan, 2016)*

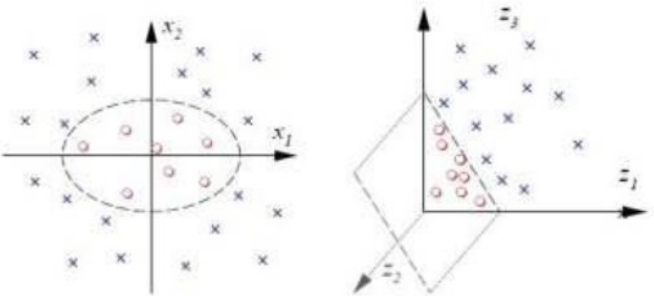
Doğrusal DVM, veri noktalarını hiper düzlemle ayırır. Şekil 4'te iki sınıflı veri setini ayıran hiper düzlemlerin geometrik gösterimi verilmiştir.



Şekil 4 : *İki sınıflı veri setini ayırabilen çeşitli hiperdüzlemler.*

(Ayhan ve Erdoğmuş, 2014).

Doğrusal olmayan DVM ise, verileri yüksek boyutlu bir uzaya taşıyarak ayrım yapar. Doğrusal olmayan iki boyutlu girdi için çok boyutlu uzaya geçişin geometrik gösterimi Şekil 5'te verilmiştir.



Şekil 5: *İki boyutlu girdi uzayının çok boyutlu bir uzaya dönüştürülmesi*

(Ayhan ve Erdoğmuş, 2014).

4.3. Programcılık ve Makine Öğrenmesinin Kesişimi

Programlama ve makine öğreniminin kesişimi, sağlık, finans ve otonom sistemler gibi birçok sektörü etkileyerek teknolojide önemli ilerlemelerin katalizörü olmuştur. Programlama, makine öğrenimi algoritmalarının uygulanması ve rafine edilmesini sağlayarak, matematiksel modellerin gerçek dünyaya uygulanabilir çözümlere dönüşmesini kolaylaştırır. Makine öğreniminde programlama, veri hazırlığı, model seçimi, algoritma parametre ayarları ve model performansının değerlendirilmesini kapsar. Bu, sadece algoritmaların yürütülmesinin ötesine geçer ve ölçeklenebilir, sürdürülebilir ve güvenli uygulamalar sağlar (Jordan ve Mitchell, 2015).

Python ve Makine Öğrenmesi Kütüphaneleri: Python, basitliği ve okunabilirliği sayesinde makine öğreniminde popülerdir. TensorFlow, PyTorch, Scikit-learn ve Pandas gibi kütüphaneler Python'u tercih edilen dil haline getirir. TensorFlow ve PyTorch, derin öğrenme için güçlü araçlar sunarken, Scikit-learn çeşitli makine öğrenimi algoritmalarının uygulanmasını sağlar. Pandas ise veri manipülasyonu ve analizi için önemlidir (Gevorkyan vd., 2019).

R ve İstatistiksel Analiz: R, istatistiksel analiz ve grafiksel gösterimler için güçlü destek sunar. Geniş paket havuzu, ekonometri, mekansal analiz ve biyoinformatik gibi alanlarda

işlevselliği artırır. R, keşifsel veri analizi, istatistiksel modelleme ve tekrarlanabilir araştırmalar için idealdir. R Markdown ve Shiny, analizlerin paylaşımını kolaylaştırır (Kolaczyk ve Csárdi, 2014; Dryden ve Mardia, 2016).

Java ve Büyük Veri Ekosistemleri: Java, büyük veri ekosistemlerinin geliştirilmesi ve yönetilmesinde baskın bir dil olarak bilinir. Apache Hadoop ve Apache Spark gibi çerçeveler, büyük veri işlemenin ölçeğini ve karmaşıklığını ele almak için kullanılır. Java, güvenli, güvenilir ve bakımı yapılabilir kodu kolaylaştırır, bu da büyük veri projelerinde çok önemlidir (Qazi ve Abbas, 2024; Nadal vd., 2019; Mehta, 2017).

C++ ve Performans Odaklı Uygulamalar: C++, sistem kaynakları üzerindeki doğrudan kontrolü ve düşük seviyeli bellek manipülasyonu olanaklarıyla, performans odaklı uygulamalarda kritik öneme sahiptir. Nesne yönelimli programlama, manuel bellek yönetimi ve STL gibi özellikler, yüksek hızlı ve verimli uygulamaların geliştirilmesini sağlar. C++, yüksek performanslı bilgi işlemede (HPC) de yaygın olarak kullanılır (Ostrowski ve Gaczkowski, 2021; Lin vd., 2022).

Makine öğrenimi (ML), yazılım geliştirme süreçlerini verimlilik ve etkinlik açısından dönüştürerek, yazılım sistemlerinin veri odaklı öğrenme ile performanslarını artırmasını sağlar. ML'nin entegrasyonu, yazılım geliştirmenin yaşam döngüsündeki gereksinim analizi, tasarım, test, dağıtım ve bakım gibi önemli aşamalarda büyük rol oynar. Gereksinim analizinde ML, büyük veri kümelerini analiz ederek eğilimleri ve kalıpları belirler, böylece yazılımın işlevselliği ve tasarımına dair önemli içgörüler sağlar. Tasarım aşamasında, ML algoritmaları önceki projelerdeki başarılı kalıplara dayalı mimariler önerir. Test aşamasında, ML modelleri test senaryosu oluşturma, yürütme ve hata analizini optimize eder (Wan vd., 2019; Amershi vd., 2019).

Makine öğrenimi, gereksinim analizinde veri odaklı yaklaşımları mümkün kılarak kullanıcı ihtiyaçlarını daha dinamik ve doğru bir şekilde haritalar. Kullanıcı geri bildirimleri ve pazar trendleri gibi büyük veri kaynaklarını işleyerek, ML algoritmaları

kalıplar ve ilişkiler ayırt eder. Duygu analizi araçları, kullanıcı geri bildirimlerini tarayarak ortak sorunları ve memnuniyet alanlarını belirler. Dinamik ve uyarlanabilir gereksinim modelleri, yeni veriler elde edildikçe güncellenir ve kullanıcı hikayelerini daha hedefe yönelik hale getirir (Maalej vd., 2015; Vogelsang ve Borg, 2019).

Makine öğrenimi, tasarım ve modelleme araçlarının entegrasyonuyla yazılım geliştirme süreçlerini iyileştirir. ML algoritmaları, geçmiş tasarım verilerini analiz ederek mevcut tasarım kararlarını bilgilendirir. Kullanıcı arayüzü (UI) tasarımında, ML kullanıcı tercihlerini ve davranışlarını tahmin ederek daha sezgisel arayüzler oluşturulmasını sağlar. ML ayrıca yazılım mimarisi tasarımında projenin özel gereksinimlerine dayalı olarak en etkili kalıpları ve mimarileri önerir (Amershi vd., 2019; Giray, 2021; Gupta vd., 2021).

Makine öğrenimi, test süreçlerinin otomasyonu ve optimizasyonunda kritik rol oynar. ML algoritmaları, geçmiş test verilerini analiz ederek potansiyel sorun alanlarını tahmin eder ve test çalışmalarını stratejik olarak yönlendirir. Test senaryosu önceliklendirmesi, en alakalı testlerin ilk olarak yürütülmesini sağlar. Kullanılabilirlik ve erişilebilirlik testlerinde ML, gerçek dünyadaki kullanım modellerini taklit eden testleri otomatik olarak oluşturur. Anormallik tespiti ve yük/performance testlerinde ML, performans verilerini analiz ederek uygulamaların kararlılığını ve güvenilirliğini artırır (Stouky vd., 2018; Deming vd., 2021; Durelli vd., 2019).

Sürekli Entegrasyon (CI) ve Sürekli Dağıtım (CD), yazılım geliştirmede çeviklik ve yanıt verebilirliği artıran temel uygulamalardır. CI, kod değişikliklerinin paylaşılan bir depoya sık sık entegre edilmesini sağlar ve her entegrasyon otomatik testlerle doğrulanır. CD, kod değişikliklerini otomatik olarak test ve üretim ortamlarına dağıtır. ML'nin entegrasyonu, derleme verilerini analiz ederek hataları öngören kalıpları belirler ve test paketlerini optimize eder. Kullanıcı davranış kalıplarını öğrenen ML modelleri, yeni özelliklerin kullanıcı katılımı üzerindeki etkisini tahmin eder (Mishra ve Sharma, 2024; Saidani vd., 2022).

Makine öğrenimi (ML) ve programlamanın entegrasyonu, sağlık, finans, ulaşım ve tüketici hizmetleri gibi birçok sektörde karmaşık görevleri yerine getirebilen akıllı sistemlerin geliştirilmesini sağlar. Sağlıkta, ML algoritmaları tıbbi görüntüleme verilerini analiz ederek hastalıkları erken tespit eder. Ayrıca, tedavi planlarını kişiselleştirmek ve sağlık hizmeti kaynaklarını yönetmek için de kullanılır (Kızılkaya ve Oğuzlar, 2018). Finans sektöründe, ML algoritmaları hisse senedi fiyatlarını tahmin eder, risk yönetimi yapar ve dolandırıcılık tespitinde yardımcı olur. Ulaşımda, ML ile entegre edilmiş programlama, otonom sürüş teknolojisini geliştirir. Tüketici hizmetlerinde, e-ticaret ve yayın platformları, müşteri verilerini analiz ederek kişiselleştirilmiş öneriler sunar (Dixon vd., 2020).

Web uygulamalarında ML, kullanıcı etkileşimini ve memnuniyetini artırmak için kişiselleştirilmiş içerik sunumu ve uyarlanabilir kullanıcı arayüzleri sağlar. ML algoritmaları, kullanıcı davranışını analiz ederek bireysel tercihleri karşılayan içerikler sunar. Örneğin, kişiselleştirilmiş haber akışları ve ürün önerileri oluşturulur. Uyarlanabilir kullanıcı arayüzleri, kullanıcı etkileşimlerine göre otomatik olarak ayarlanır. Ayrıca, ML destekli sohbet robotları ve sanal asistanlar, kullanıcı sorgularını doğal dilde yorumlayarak yanıtlar (Winter ve Jackson, 2020).

ML, mobil uygulamalarda kullanıcı davranışını analiz ederek kullanıcı etkileşimini ve deneyimini optimize eder. Kullanıcı tıklamaları, oturum süresi, uygulama içi etkileşimler gibi veri noktaları toplanır ve işlenir. Kullanıcı segmentleri belirlenir ve hedefli pazarlama stratejileri uygulanır. Tahmine dayalı analitik ile gelecekteki kullanıcı eylemleri tahmin edilir. ML, ayrıca mobil uygulamaların güvenliğini artıran anomali tespit algoritmaları sağlar (Jiang vd., 2021).

YZ, oyun geliştirmede önemli bir rol oynar. Akıllı NPC'ler, prosedürel içerik üretimi ve oyun testi gibi alanlarda kullanılır. NPC'ler, oyuncu eylemlerine tepki veren karmaşık davranışlar sergiler. Prosedürel içerik üretimi, oyun içeriğini otomatik olarak oluşturur ve tekrar oynanabilirliği artırır. YZ, oyun testi ve

dengelemede kullanılarak hataları tespit eder ve oyun dengesini sağlar (Yakan, 2022).

Robotik sistemler ve otomasyon, üretim süreçlerinde verimliliği artırır ve insan hatasını azaltır. Üretim hatlarında kullanılan robotlar, tekrarlayan görevleri yüksek hız ve doğrulukla gerçekleştirir. Sağlıkta, robotik cerrahi sistemler ve otomatik teşhis araçları kullanılır. Tarımda, dronlar ve otonom kara araçları ekim, hasat gibi görevleri yerine getirir. Lojistikte, AGV'ler ve AMR'ler depo operasyonlarını otomatikleştirir. Yaratıcı endüstrilerde, robotik kamera sistemleri ve sanat yapımında kullanılan robotlar önemli yenilikler sunar (Barnes vd., 2021).

ML ve programlama, robotik sistemlerin uyarlanabilirliğini ve karar verme yeteneklerini artırır. Bu sistemlerin etik kullanımı, işten çıkarma, mahremiyet ve güvenlik gibi konuları dikkate alarak dikkatli düzenleme ve etik programlama uygulamaları gerektirir (Afzal vd., 2020).

3.4. Uygulama ve Vaka Çalışmaları

Makine öğrenimi, yapay zekâ ve sofistike programlama teknolojileri, birçok sektörde akıllı sistemlerin ve otomatik süreçlerin geliştirilmesine olanak tanımıştır. Bu entegrasyon, sağlık, finans, ulaşım ve eğlence gibi alanlarda verimliliği artırmak, karar verme süreçlerini optimize etmek ve kişiselleştirilmiş kullanıcı deneyimleri sunmak için kullanılır.

Makine öğrenimi algoritmaları, tıbbi görüntüleme verilerini analiz ederek hastalıkları erken tespit etmeye yardımcı olur. Derin öğrenme modelleri, kanser ve diyabetik retinopati gibi hastalıkların teşhisinde kullanılır. Ayrıca, tahmine dayalı analitik, hasta sonuçlarını tahmin etmek ve tedavi rejimlerini kişiselleştirmek için kullanılır (Atalay ve Çelik, 2017).

Algoritmik ticaret, yüksek hızda işlemler gerçekleştirerek piyasa verimliliğini artırır. Bu sistemler, büyük veri kümelerini analiz ederek ticaret fırsatlarını belirler ve otomatik olarak emirler verir. Yüksek frekanslı ticaret (HFT) sistemleri, işlemleri mikrosaniyeler içinde gerçekleştirir. Ayrıca, piyasa yapıcılığı

stratejileri ve risk yönetimi algoritmaları, finansal piyasaların likiditesini ve stabilitesini sağlar (Theate ve Ernst, 2021; Massei, 2023).

Otonom araçlar ve akıllı trafik yönetimi sistemleri, güvenliği ve verimliliği artırmak için makine öğrenimi algoritmaları kullanır. Bu sistemler, trafik akışını optimize eder ve çevrelerinden öğrenerek karar verme yeteneklerini geliştirir (Mahalakshmi vd., 2022).

Kişiselleştirilmiş içerik öneri sistemleri, kullanıcı davranışını ve tercihlerini analiz ederek içerik önerir. Bu algoritmalar, kullanıcı etkileşimini artırarak daha yüksek elde tutma oranları sağlar (Meena vd., 2020).

Algoritmik ticaret, finansal işlemlerin hızını, hacmini ve karmaşıklığını artırarak piyasa verimliliğini artırır. Bu sistemler, geçmiş verilere dayanarak ticaret stratejileri geliştirir ve yüksek hızda emirler gerçekleştirir. HFT sistemleri, mikrosaniyeler içinde ticaret yaparak fiyat farklılıklarından yararlanır. Ayrıca, risk yönetimi algoritmaları, finansal piyasalardaki çeşitli riskleri izler ve kontrol eder (Bandi ve Kothari, 2022).

Makine öğrenimi modelleri, tıbbi görüntüleme ve elektronik sağlık kayıtları gibi verileri analiz ederek hastalıkları tespit eder ve hasta sonuçlarını tahmin eder. Bu modeller, kişiselleştirilmiş tıp stratejileri geliştirir ve tedavi planlarını optimize eder. Robotik cerrahi sistemler, cerrahi prosedürlerin hassasiyetini artırarak iyileşme sürelerini kısaltır (Khanna vd., 2022; Ahmed vd., 2020).

Makine öğrenimi algoritmaları, müşteri etkileşimlerinden elde edilen verileri analiz ederek kişiselleştirilmiş alışveriş deneyimleri sunar. Müşteri segmentasyonu ve tahmine dayalı analitik, pazarlama stratejilerini optimize eder ve envanter yönetimini iyileştirir. Dinamik fiyatlandırma algoritmaları, piyasa koşullarına hızlı ayarlamalar yaparak rekabet avantajı sağlar (Güven ve Güven, 2023; Li, 2023).

Yapay zekâ, trafik yönetimi, enerji yönetimi, kamu güvenliği ve çevresel izleme gibi alanlarda akıllı şehirlerin etkinliğini artırır. Trafik akışını optimize eden yapay zeka sistemleri, enerji talebini

tahmin eden akıllı şebekeler ve çevresel riskleri yönetmek için sensör verilerini analiz eden sistemler kullanılır. Bu uygulamalar, kentsel yaşam kalitesini artırır ve sürdürülebilirliği sağlar (Allam ve Dhunny, 2019; Ullah vd., 2020; Herath ve Mittal, 2022).

4.Tartışma ve Sonuç

Makine öğrenmesi, sağlık ve finans gibi alanlarda büyük fırsatlar sunarken, algoritmik önyargılar ve veri gizliliği gibi etik zorluklar da yaratmaktadır. Bu çalışma, algoritmalardaki yeniliklerin ve uygulama alanlarının derinlemesine incelenmesiyle, bu zorlukların üstesinden gelme yollarını tartışmaktadır. Bir yandan algoritmaların karmaşıklığı arttıkça, modelin şeffaflığı ve yorumlanabilirliği konusunda endişeler de artmaktadır. Yapay zekâ sistemlerinin karar verme süreçlerinin insanlar tarafından anlaşılabilir olması gerektiği vurgulanmaktadır. Bu, özellikle yüksek risk taşıyan uygulama alanlarında (örneğin, sağlık hizmetleri veya yargı sistemleri) yanlış kararların ciddi sonuçlara yol açabileceği durumlarda kritik öneme sahiptir.

Öte yandan veriye dayalı modellerin eğitiminde kullanılan verilerin kalitesi, algoritmanın genel performansını doğrudan etkilemektedir. Veri setlerindeki eksiklikler, yanlılık ve düşük kalite, modelin yanıltıcı sonuçlar üretmesine neden olabilir. Bu nedenle, sağlam veri yönetimi ve ön işleme stratejilerinin geliştirilmesi, model doğruluğunu artırma potansiyeline sahiptir. Ayrıca, makine öğrenmesi uygulamalarının sosyal ve etik boyutları üzerinde durulmaktadır. Teknolojinin adil ve etik kullanımını sağlamak amacıyla, algoritmik önyargıları azaltacak yöntemlerin geliştirilmesi gerekmektedir. Bu, hem algoritma tasarım sürecinde çeşitliliği ve kapsayıcılığı artırmak hem de etik denetim mekanizmalarını güçlendirmek yoluyla sağlanabilir.

Makine öğrenmesinin geleceği, teknik yeteneklerin yanı sıra etik standartlar ve düzenleyici çerçevelerle de şekillenecektir. Bu alanda yapılan araştırmaların hem teknolojik yenilikleri hem de toplumsal değerleri göz önünde bulundurmasının, sürdürülebilir ve adil bir teknolojik ilerleme sağlama potansiyelini artıracağı düşünülmektedir.

Makine öğrenmesi alanındaki çalışmalar, algoritmalara ve modellere dayalı bir dizi çözüm sunarak bilimsel ve teknolojik ilerlemeyi şekillendirmeye devam etmektedir. Bu çalışmada tartışılan temel konular, algoritmalardan uygulama alanlarına, yöntemlerin sınıflandırılmasından karşılaşılan zorluklara kadar geniş bir yelpazeyi kapsamaktadır. Gözetimli, gözetimsiz ve pekiştirmeli öğrenme gibi farklı kategorilere ayrılan yaklaşımlar, makine öğrenmesinin çeşitli problemlere esnek çözümler sunma kapasitesini göstermektedir. Bununla birlikte, algoritmaların etkinliğini sınırlayan veri kalitesi, önyargılar, hesaplama maliyetleri ve genelleştirme yeteneği gibi zorluklar, araştırmacılar için önemli geliştirme alanları olmaya devam etmektedir. Gelecekte bu zorlukların üstesinden gelmek için daha sofistike algoritmaların ve modellerin geliştirilmesi beklenmektedir. Ayrıca, etik standartların ve düzenlemelerin, teknolojik yeniliklerle uyumlu bir şekilde gelişmesi gerekmektedir. Bu, makine öğrenmesinin toplum üzerindeki faydalı etkilerini maksimize ederken, potansiyel olumsuz etkilerini en aza indirmek için büyük bir öneme sahiptir.

Sonuç olarak makine öğrenmesi, potansiyelini tam olarak ortaya koyabilmesi için multidisipliner bir yaklaşım gerektiren, dinamik ve sürekli gelişen bir alandır. Akademik ve sektörel araştırmaların bu alanda sürekli olarak yenilikler yapması, makine öğrenmesinin geleceğini şekillendirme potansiyeline işaret etmektedir. Önümüzdeki dönemde makine öğrenmesinin daha da entegre ve etik bir teknoloji haline gelmesinin büyük önem arz edeceği düşünülmektedir.

Kaynaklar

Afzal, A., Le Goues, C., Hilton, M., & Timperley, C. S. (2020, October). A study on challenges of testing robotic systems. In 2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST) (pp. 96-107). IEEE.

Ahmed, Z., Mohamed, K., Zeeshan, S., & Dong, X. (2020). Artificial intelligence with multi-functional machine learning platform development for better healthcare and precision medicine. Database, 2020, baaa010.

Akman, M. (2010). Veri madenciliğine genel bakış ve random forests yönteminin incelenmesi: sağlık alanında bir uygulama. (Yüksek Lisans Tezi). Ankara Üniversitesi, Sağlık Bilimleri Enstitüsü, Ankara.

Alpaydın, E. (2004). Introduction to Machine Learning. The MIT Press, Cambridge.

Allam, Z., & Dhunny, Z. A. (2019). On big data, artificial intelligence and smart cities. Cities, 89, 80-91.

Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., ... & Zimmermann, T. (2019, May). Software engineering for machine learning: A case study. In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP) (pp. 291-300). IEEE.

Atalay, M., & Çelik, E. (2017). Büyük veri analizinde yapay zekâ ve makine öğrenmesi uygulamalari-artificial intelligence and machine learning applications in big data analysis. Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 9(22), 155-172.

Ayhan, S., & Erdoğmuş, Ş. (2014). Destek vektör makineleriyle sınıflandırma problemlerinin çözümü için çekirdek fonksiyon seçimi. Eskişehir Osmangazi Üniversitesi İibf Dergisi, 9(1), 175-198.

Bandi, S., & Kothari, A. (2022). Artificial intelligence: An asset for the financial sector. Impact of Artificial Intelligence on Organizational Transformation, 259-287.

Barnes, E., Morgan, G., Hake, K., Devine, J., Kurtz, R., Ibendahl, G., ... & Holt, G. (2021). Opportunities for robotic systems and automation in cotton production. AgriEngineering, 3(2), 339-362.

Berrar, D. (2019). Bayes' Theorem and Naive Bayes Classifier.

Breiman, L., & Cutler, A. Random Forest. University of Stat Barkeley Ağ Sitesi.

Ceryan, N. (2016). En Küçük Kareler Destek Vektör Makineleri (LS-SVM) Kullanarak Kaya Malzemesi Tanjant Elastisite Modülünün Tahmini. Hacettepe Üniversitesi Yerbilimleri Uygulama ve Araştırma Merkezi Bülteni, 37(2), 121-138.

Deming, C., Khair, M. A., Mallipeddi, S. R., & Varghese, A. (2021). Software Testing in the Era of AI: Leveraging Machine Learning and Automation for Efficient Quality Assurance. Asian Journal of Applied Science and Engineering, 10(1), 66-76.

Demircioğlu, H. Z., & Bilge, H. Ş. (2015). Yumurtalık kanseri veri kümesindeki gen ifadelerinin veri madenciliği ile analizi. Marmara Fen Bilimleri Dergisi, 27(4), 125-134. Doi: 10.7240/mufbed.89154

Dixon, M. F., Halperin, I., & Bilokon, P. (2020). Machine learning in finance (Vol. 1170). New York, NY, USA: Springer International Publishing.

Dryden, I. L., & Mardia, K. V. (2016). Statistical shape analysis: with applications in R (Vol. 995). John Wiley & Sons.

Durelli, V. H., Durelli, R. S., Borges, S. S., Endo, A. T., Eler, M. M., Dias, D. R., & Guimarães, M. P. (2019). Machine learning applied to software testing: A systematic mapping study. IEEE Transactions on Reliability, 68(3), 1189-1212.

Emir, Ş. (2013). Yapay sinir ağları ve destek vektör makinelerinin sınıflandırma performanslarının karşılaştırılması: borsa endeks yönünün tahmini üzerine bir uygulama. (Doktora Tezi). İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul.

Gevorkyan, M. N., Demidova, A. V., Demidova, T. S., & Sobolev, A. A. (2019). Review and comparative analysis of machine learning libraries for machine learning. Discrete and Continuous Models and Applied Computational Science, 27(4), 305-315.

Giray, G. (2021). A software engineering perspective on engineering machine learning systems: State of the art and challenges. Journal of Systems and Software, 180, 111031.

Gupta, R., Srivastava, D., Sahu, M., Tiwari, S., Ambasta, R. K., & Kumar, P. (2021). Artificial intelligence to deep learning: machine intelligence approach for drug discovery. Molecular diversity, 25, 1315-1360.

Güven, H., & Güven, E. T. A. (2023). Yapay Zekâ Uygulamalarının E-Ticarette Kullanımı. International Journal of Management and Administration, 7(13), 69-94.

Herath, H. M. K. K. M. B., & Mittal, M. (2022). Adoption of artificial intelligence in smart cities: A comprehensive review. International Journal of Information Management Data Insights, 2(1), 100076.

Jiang, H., He, M., Xi, Y., & Zeng, J. (2021). Machine-learning-based user position prediction and behavior analysis for location services. Information, 12(5), 180.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260.

Kartal, E. (2015). Sınıflandırmaya dayalı makine öğrenmesi teknikleri ve kardiyolojik risk değerlendirmesine ilişkin bir uygulama. (Doktora Tezi). İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Khanna, N. N., Maindarkar, M. A., Viswanathan, V., Fernandes, J. F. E., Paul, S., Bhagawati, M., ... & Suri, J. S. (2022,

December). Economics of artificial intelligence in healthcare: diagnosis vs. treatment. In Healthcare (Vol. 10, No. 12, p. 2493). MDPI.

Kızılkaya, Y. M., & Oğuzlar, A. (2018). Bazı Denetimli Öğrenme Algoritmalarının R Programlama Dili İle Kıyaslanması. Karadeniz Uluslararası Bilimsel Dergi, 37(37), 90-98.

Kolaczyk, E. D., & Csárdi, G. (2014). Statistical analysis of network data with R (Vol. 65). New York: Springer.

Li, L. (2023). Analysis of e-commerce customers' shopping behavior based on data mining and machine learning. Soft Computing, 1-10.

Lin, W. C., Deakin, T., & McIntosh-Smith, S. (2022, November). Evaluating ISO C++ Parallel Algorithms on Heterogeneous HPC Systems. In 2022 IEEE/ACM International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS) (pp. 36-47). IEEE.

Maalej, W., Nayebi, M., Johann, T., & Ruhe, G. (2015). Toward data-driven requirements engineering. IEEE software, 33(1), 48-54.

Mahalakshmi, V., Kulkarni, N., Kumar, K. P., Kumar, K. S., Sree, D. N., & Durga, S. (2022). The role of implementing artificial intelligence and machine learning technologies in the financial services industry for creating competitive intelligence. Materials Today: Proceedings, 56, 2252-2255.

Massei, G. (2023). Algorithmic Trading: An Overview and Evaluation of Its Impact on Financial Markets.

Meena, M. R., Jingar, M. P., & Gupta, S. (2020). Artificial intelligence: A digital transformation tool in entertainment and media industry. Our Heritage, 68(1), 4661-4675.

Mehta, R. (2017). Big Data Analytics with Java. Packt Publishing Ltd.

Mijwel, M. M., (2017). Yapay Sinir Ağlar Yapısı ve Fonksiyonu. DOI:10.13140/RG.2.2.32068.14723.

Mishra, A., & Sharma, A. (2024). Deep learning based continuous integration and continuous delivery software defect prediction with effective optimization strategy. Knowledge-Based Systems, 296, 111835.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). Foundations of Machine Learning. The MIT Press, Cambridge.

Nadal, S., Romero, O., Abelló, A., Vassiliadis, P., & Vansummeren, S. (2019). An integration-oriented ontology to govern evolution in big data ecosystems. Information systems, 79, 3-19.

Ostrowski, A., & Gaczkowski, P. (2021). Software Architecture with C++: Design modern systems using effective architecture concepts, design patterns, and techniques with C++ 20. Packt Publishing Ltd.

Öztemel, E. (2012). Yapay Sinir Ağları. Papatya Yayıncılık, İstanbul.

Öztürk, K., Şahin M. E., (2018). Yapay Sinir Ağları ve Yapay Zeka'ya Genel Bir Bakış. Takvim-i Vekayi, 6(2), 25-36.

Özgür, A., & Erdem, H. (2012). Saldırı tespit sistemlerinde kullanılan kolay erişilen makine öğrenme algoritmalarının karşılaştırılması. Bilişim Teknolojileri Dergisi, 5(2), 41-48.

Qazi, A. A., & Abbas, E. (2024). Big Data and Java are integrated with machine learning. International Journal of Multidisciplinary Sciences and Arts, 3(1), 289-297.

Saidani, I., Ouni, A., & Mkaouer, M. W. (2022). Improving the prediction of continuous integration build failures using deep learning. Automated Software Engineering, 29(1), 21.

Shearer, C. (2000). The CRISP-DM model: the new blueprint for data mining. Journal of data warehousing, 5(4), 13–22.

Stouky, A., Jaoujane, B., Daoudi, R., & Chaoui, H. (2018). Improving software automation testing using jenkins, and machine learning under big data. In Big Data Technologies and Applications: 8th International Conference, BDTA 2017, Gwangju, South Korea, November 23–24, 2017, Proceedings 8 (pp. 87-96). Springer International Publishing.

Theate, T., & Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. Expert Systems with Applications, 173, 114632.

Türk Dil Kurumu, 2024, Öğrenmek, http://www.tdk.gov.tr/

Ullah, Z., Al-Turjman, F., Mostarda, L., & Gagliardi, R. (2020). Applications of artificial intelligence and machine learning in smart cities. Computer Communications, 154, 313-323.

Vogelsang, A., & Borg, M. (2019, September). Requirements engineering for machine learning: Perspectives from data scientists. In 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW) (pp. 245-251). IEEE.

Wan, Z., Xia, X., Lo, D., & Murphy, G. C. (2019). How does machine learning change software development practices?. IEEE Transactions on Software Engineering, 47(9), 1857-1871.

Winter, M., & Jackson, P. (2020, July). Flatpack ML: How to support designers in creating a new generation of customizable machine learning applications. In International Conference on Human-Computer Interaction (pp. 175-193). Cham: Springer International Publishing.

Yang, F. J. (2018, December). An implementation of naive bayes classifier. In 2018 International conference on computational science and computational intelligence (CSCI) (pp. 301-306). IEEE.

Yakan, S. A. (2022). Analysis of development of artificial intelligence in the game industry. International Journal of Cyber and IT Service Management, 2(2), 111-116.

BÖLÜM X

Daha Yüksek Verimlilik ve Güvenlik için Açık Bankacılık Platformu Geliştirme

Zeynep YAVUZ[1]
Zeynep Nur SANDIKÇI[2]
Önder DEMİR[3]
Kazım YILDIZ[4]

GİRİŞ

Bankacılık, bankaların yürüttüğü tüm faaliyetleri kapsayan bir terimdir. Bankacılık, mevduat kabul etme, kredi sağlama, ödeme hizmetleri sunma, döviz işlemleri yapma ve finansal danışmanlık hizmetleri verme gibi işlemleri içerir (Saunders & Cornett, 2019). Fintek (Finansal Teknolojiler) ise, finansal hizmetlerin teknolojik araçlar ve yenilikler kullanılarak sunulmasını sağlayan bir ifadedir

[1] Sipay Electronic Money and Payment Services Inc, Istanbul, Turkey, 0000-0003-0464-4172:
[2] Sipay Electronic Money and Payment Services Inc, Istanbul, Turkey, 0000-0001-9201-3690:
[3] Computer Engineering, Faculty of Technology. Marmara University, 34854, Istanbul, Turkey, 0000-0003-4540-663X:
[4] Computer Engineering, Faculty of Technology. Marmara University, 34854, Istanbul, Turkey, 0000-0001-6999-1410:

(Schueffel, 2016). Teknolojinin hızlı ilerleyişi, finans sektörü de dahil birçok alanda, nitelikli hizmet sunmak ve dijital dönüşüme ayak uydurmak adına yenilikçi hamleler yapma gerekliliğini doğurmuştur. Türkiye, bankacılık ve fintek sektörü açısından büyük bir potansiyele sahip bir ülkedir (Demirez, Gür, & Yaşa Özeltürkay, 2021) . Genç, dinamik ve teknolojiye yatkın bir nüfus yapısına sahip olan Türkiye, aynı zamanda stratejik konumu ve büyüyen ekonomisiyle bu sektörlerin gelişmesi için uygun bir zemin sunmaktadır. Bu faktörler, bankacılık ve fintek sektörlerinin hızlı bir şekilde büyümesine ve küresel pazarda rekabetçi bir konuma gelmesine olanak tanımaktadır. Geleneksel bankacılık hizmetlerinin yanı sıra, dijital bankacılık uygulamaları ve mobil bankacılık çözümleri de hızla yaygınlaşmaktadır. Özellikle, büyük bankalar müşterilerine sundukları ürün ve hizmet yelpazesini genişleterek, kredi kartlarından bireysel kredilere, yatırım hizmetlerinden sigorta çözümlerine kadar geniş bir finansal portföy sunmaktadır. Bankacılık sektöründeki rekabet, müşteri memnuniyetini artırmak ve yenilikçi çözümler sunmak adına bankaları dijitalleşme ve teknolojik yatırımlar yapmaya zorlamaktadır. Fintek şirketleri, ödeme sistemlerinden sigorta teknolojilerine, dijital bankacılıktan varlık yönetimine kadar çeşitli finansal hizmetler sunmaktadır. Türkiye’nin genç nüfusu ve dijitalleşme oranının yüksek olması, fintek ekosisteminin büyümesini hızlandıran en önemli faktörlerden biridir. Bunun yanında, Türkiye'deki fintek girişimleri, küresel pazarlarda da ses getirmekte ve yatırımcıların ilgisini çekmektedir.

Açık bankacılık kavramı, kullanıcıların müşterilerin birden fazla hesabını tek bir platform üzerinden takip etmelerini sağlayan yenilikçi bir finansal yaklaşımı ifade etmektedir. Finans alanında dijitalleşmenin getirdiği kolaylık ve esneklik, açık bankacılık ile daha fazla gelişmiştir Gelecekte bankacılık sektöründe yaygın olacağı öngörülen Açık Bankacılık kavramı, fintek girişimleri için önemli fırsatlar sunmakta ve yeni iş alanlarının önünü açmaktadır.

Açık bankacılık, önemli yararlar sağlar: müşteri deneyimini geliştirir, yeni gelir kaynakları oluşturur ve geleneksel olarak yeterince hizmet verilmeyen pazarlar için uzun vadeli bir iş modeli sunar (McKinsey&Company, 2017).

Günümüzde, finansal işlemlerimizle ilgili tüm bilgiler bankalar tarafından kayıt altına alınmaktadır. Otomatik ödeme talimatları, kredi kartı harcamaları, alışverişler ve borç durumları gibi bilgiler bankaların veri tabanlarında saklanır. Açık bankacılık platformu, bu verilerin kontrolünü tekrar bireylere vermeyi amaçlamaktadır. Bu sistem, bankaların ellerinde bulunan kişisel finansal verilerin, kullanıcıların onayı ile üçüncü taraf mobil uygulamalar ve internet hizmetleri tarafından erişilebilir olmasını sağlar. Bu nedenle, kullanıcılar hangi uygulamanın ya da hizmetin bu verilere erişebileceğine kendileri karar verir ve veriler üzerinde daha fazla kontrol sahibi olurlar. Bu yöntem, bireylerin finansal verilerini diledikleri şekilde yönetebilmesine olanak tanır ve bankaların veri tekelini azaltır (Fidanboy, 2022).

Bu çalışma, bireysel ve kurumsal müşterilere yönelik bankacılık hizmetlerini tek bir açık bankacılık platformunda birleştirmeyi hedeflemektedir. Platform, kullanıcıların çeşitli banka hesaplarını aynı arayüz üzerinden yönetmesini ve para transferi, kredi başvurusu gibi işlemleri kolaylaştırmasını sağlar. İşlemleri otomatikleştirerek hataları ve operasyonel yükü azaltır. Global standartlara uygun olarak tasarlanan platform, gelecekteki genişlemeler ve uluslararası entegrasyonlar için ölçeklenebilir ve modüler bir yapıya sahiptir. Çalışma, açık bankacılığın finansal şeffaflık ve verimlilik potansiyelini vurgularken, veri güvenliği ve müşteri gizliliği konularına stratejik çözümler önermektedir.

1.Türkiye’de Bankacılık ve Fintek Sektörü

Türkiye, bankacılık ve fintek sektörü açısından önemli bir potansiyele sahiptir. Özellikle genç ve dinamik bir nüfus altyapısı ile yüksek teknoloji benimseme oranına ve büyüyen bir ekonomiye sahiptir. Bu faktörler, bankacılık ve fintek sektörünün büyümesine katkıda bulunmaktadır. Türkiye'de bankacılık sektörü, güçlü bir temele sahiptir. Türkiye Bankalar Birliği Haziran 2024 raporuna göre aktif banka sayısı 63 olarak ifade edilmiştir (Türkiye Bankalar Birliği, Banka, Çalışan ve Şube Sayıları, 2024). Bu bankalar, geniş bir müşteri kitlesine ve geniş bir ürün ve hizmet yelpazesine sahiptir. Türkiye'de fintek sektörü, hızla büyüyen bir sektördür. 2022 yılı itibarıyla, Türkiye'de 696 aktif fintek şirketi bulunmaktadır (T.C. Cumhurbaşkanlığı Finans Ofisi, 2024). Bu şirketler, çeşitli finansal hizmetler sunmaktadır.

Türkiye’de bankacılık ve finans teknolojileri sektörünün gelişimini destekleyen çeşitli faktörler bulunmaktadır:

Yüksek teknoloji benimseme oranı: Türkiye, yüksek teknoloji benimseme oranına sahip bir ülkedir. Bu, fintek şirketlerinin yenilikçi ürünleri ve hizmetleri geliştirmesine yardımcı olmaktadır.

Büyüyen ekonomi: Türkiye, büyüyen bir ekonomiye sahiptir. Bu, bankacılık ve fintek sektörünün büyümesine olanak tanımaktadır.

Rekabet ortamı: Türkiye'de bankacılık ve fintek sektöründe rekabet oldukça yüksektir. Bu, sektörde yenilik ve gelişmeyi teşvik etmektedir.

Türkiye'de bankacılık ve fintek sektörünün gelecekte daha da büyümesi beklenmektedir. Bu büyüme, sektörde yeni iş fırsatları yaratacak ve müşterilerin daha iyi finansal hizmetlere erişmesini sağlayacaktır (Sezal, 2021).

Türkiye’de bankacılık ve finans teknolojileri sektöründeki bazı dikkate değer gelişmeler şunlardır:

Dijital bankacılık: Türkiye'de dijital bankacılık, hızla yaygınlaşmaktadır. 2022 yılı itibarıyla, Türkiye'de 10 milyondan fazla dijital bankacılık müşterisi bulunmaktadır (Türkiye Bankalar Birliği, Dijital, İnternet ve Mobil Bankacılık İstatistikleri, 2022).

Açık bankacılık: Türkiye'de açık bankacılık, 2021 yılında yürürlüğe girdi. Açık bankacılık, müşterilerin finansal verilerini üçüncü taraf kuruluşlarla paylaşmasına olanak tanımaktadır. Bu, yeni finansal hizmetler ve ürünlerin geliştirilmesini sağlayacaktır.

Fintek şirketlerinin büyümesi: Türkiye'de fintek şirketleri, hızla büyümektedir. 2023 yılı itibarıyla, Türkiye'de 696 aktif fintek şirketi bulunmaktadır. Bu şirketler, çeşitli finansal hizmetler sunmaktadır (T.C. Cumhurbaşkanlığı Finans Ofisi, 2024).

2023 Türkiye e-ticaret ekosistemi raporuna göre 2023 yılı itibarıyla e-ticaret sektörü %132 oranında büyüme kaydederek 1.855 milyar TL seviyesine yükselmiştir. Ayrıca 2020-2021 ve 2021-2022 yılları arasındaki dönemde e-ticaret işlem sayıları sırasıyla %45 ve %43 oranında artış göstermişken, 2023 yılında bu oran %23 olarak gerçekleşmiştir. 2023 yılı ile birlikte Türkiye’de 6 milyar civarında e-ticaret işlemi gerçekleşmiştir (iyzico, Dogma Alares, & ETİD, 2024).

Ülkemiz bankacılık sektörü, dijitalleşme ve hizmet kalitesinde dünyanın önde gelenleri arasında kabul edilmektedir. Teknolojinin gelişimine de bağlı tüm işlemlerin dijital platformlar üzerinden sağlanması, hemen hemen tüm banka ve finans kurumlarının kısa ve orta vadeli planları arasında yer almaktadır (Bilgel & Aksoy, 2019). Söz konusu hedefe ulaşmak için gerçekleştirilen her dijital hizmet güvenlik, mevzuat ve kurum içi süreçler göz önüne alınarak hayata geçirilmektedir. Her geçen gün

büyüyen bankacılık sektörü, dijital hizmetlerin kalitelerini artırıcı veya yeni hizmetlerin hayata geçirilmesi için yoğun çaba harcamaktadır.

2. Açık Bankacılık Platformu

Açık bankacılık, finansal hizmet sağlayıcılarının müşterilerin onayıyla bankaların ve diğer finansal kuruluşların sahip olduğu müşteri verilerini üçüncü taraflarla paylaşmasını sağlayan bir sistemdir (Uşaklı, 2021). Bu sistem, genellikle API'ler (Application Programming Interfaces - Uygulama Programlama Arayüzleri) aracılığıyla gerçekleştirilir ve finansal hizmetlerde yenilikçiliği, rekabeti ve müşteri deneyimini artırmayı hedefler.

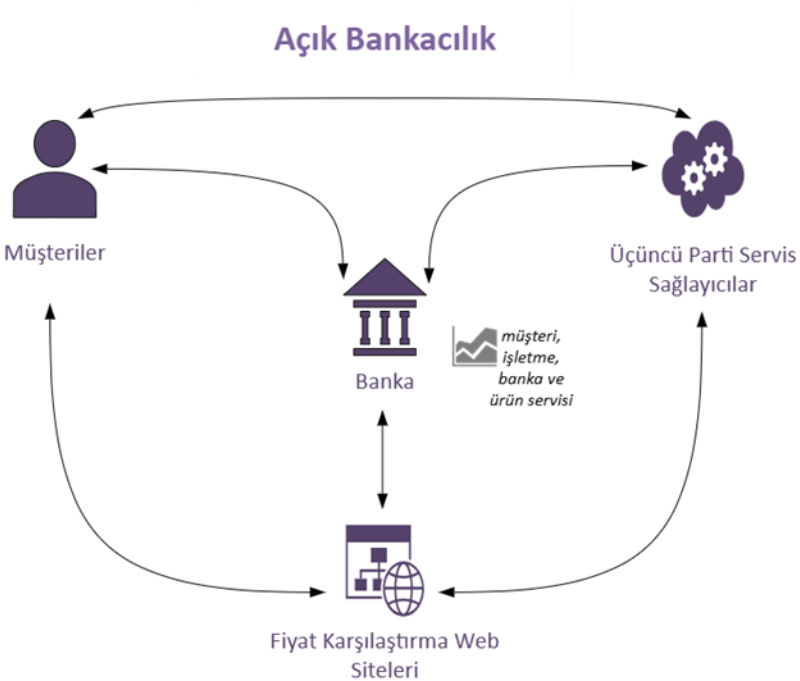
Türkiye Cumhuriyeti Merkez Bankası tarafından yasal düzenleme çerçevesinde hayata geçen Açık Bankacılık; vadesiz TL ve döviz hesaplarının, farklı banka ve lisanslı finansal kuruluşlar tarafından sunulan dijital platformlar üzerinden izlenebilmesini ve bu hesaplar arasında para transferi yapılabilmesini sağlayan modern bir bankacılık hizmet modelidir (Köseoğlu, 2023).

Açık bankacılık, temel olarak ödeme başlatma ve hesap bilgi servisleri olmak üzere iki ana hizmete üzerindedir. Hesap bilgi servisleri, üçüncü taraf kuruluşların müşterilerin hesap bilgilerine erişmesini sağlar. Bu bilgiler, bakiye, hesap hareketleri, kredi kartı bilgileri gibi finansal verileri kapsayabilir. Ödeme başlatma servisleri ise, üçüncü taraf kuruluşların müşterilerin hesaplarından para transferi yapmasını sağlar. Bu sayede, müşteriler kendi bankaları dışındaki hesaplardan para transferi yapabilirler.

Rekabet ve iş birliğini bir araya getiren bir yaklaşım olarak finansal hizmet sunan şirketler, diğer kuruluşlarla daha fazla ortak alan oluşturarak sundukları hizmetlerin çeşitliliğini genişletmektedir. Bu hizmetlerin, diğer işletmelerin altyapılarıyla bütünleşmiş şekilde API’ler etrafında yapılandırıldığı bir

ekosistemde, oyuncular ya kendi uzmanlıklarını kullanarak ya da iş birliği yaparak en iyi hizmeti hedeflemektedir.

Bankacılık sektöründe, küresel düzeyde geçerli olan sıkı düzenlemeler ve yüksek sermaye yeterliliği gereklilikleri, sektöre katılan kuruluşların sayısını önemli ölçüde sınırlamaktadır. Buna rağmen, banka dışı finansal hizmet sağlayıcılar, API(Application Programming Interface, Uygulama Programlama Arayüzü)'ler sayesinde bu ekosisteme dahil olabilmektedir. API'ler, finansal hizmet sunucuların sistemleri arasında bir köprü işlevi görmektedir. Teknik altyapıları sayesinde, müşterinin kullandığı bir uygulamanın, ihtiyaç duyulduğunda diğer uygulamalarla iletişim kurmasına olanak sağlamaktadır. API'ler aracılığıyla, finansal hizmet sunan şirketler bireylerin dijital varlıklarını analiz etme ve bu verileri kredi notu hesaplamalarında bir araç olarak kullanma imkanına sahip olmaktadır. Öte yandan, bireyler de finansal kaynaklarını daha etkili bir şekilde yönetme ve tüm varlıklarını tek bir platform üzerinden kolayca görüntüleme fırsatına kavuşmaktadır (Fintech Istanbul, 2017).



Şekil 1. Açık Bankacılık (Deloitte, 2017)

Açık bankacılık çeşitli alanlarda kullanılabilen bir sistem olup, en yaygın kullanım alanları şunlardır:

Hesap Birleştirme ve Finans Yönetimi: Açık bankacılık, kişisel finans yönetimini kolaylaştırmak için kullanılabilir. Müşteriler, tüm finansal verilerini tek bir platformdan görüntüleyerek harcamalarını takip edebilir, yatırımlarını yönetebilir ve bütçe yapabilirler.

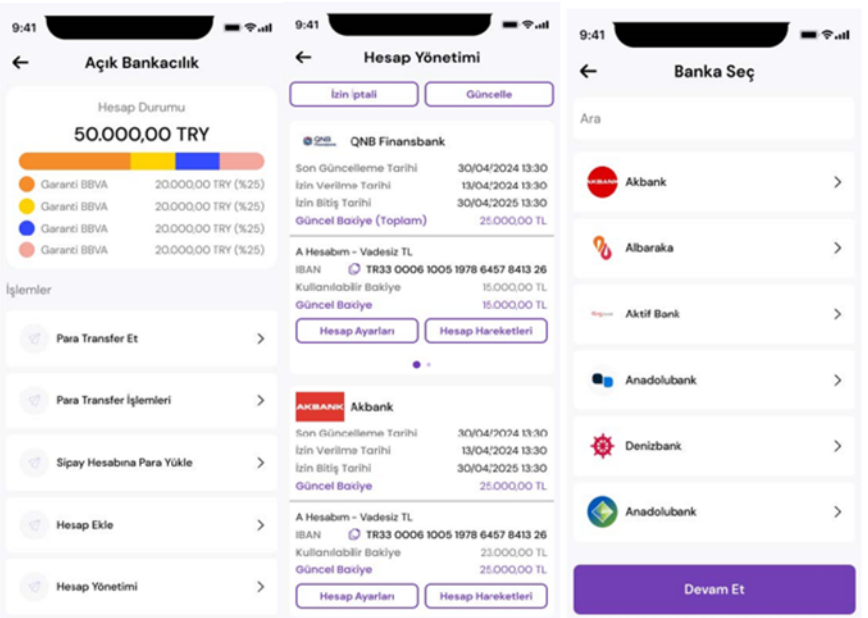
Anında Ödemeler ve Fon Transferleri: Açık Bankacılık, Birleşik Ödeme Arayüzü (UPI) platformu aracılığıyla gerçek zamanlı ödemeler ve fon transferlerini kolaylaştırır. Kullanıcılar, UPI özellikli uygulamaları kullanarak kişiden kişiye anında transferler, satıcı ödemeler ve fatura ödemelerin doğrudan banka hesaplarından başlatabilir. Sorunsuz ve kullanışlı bir ödeme deneyimi sağlar.

Kredi Puanlama ve Krediye Erişim: Kullanıcıların finansal geçmişlerini analiz ederek daha isabetli puanlama ve kredi teklifleri sunar.

İşletme Bankacılığı ve Muhasebe Entegrasyonu: Açık Bankacılık, işletmeler için bankacılık hizmetleri ile muhasebe yazılımı arasında kusursuz entegrasyonu kolaylaştırabilir. İşletmelerin finansal süreçleri otomatikleştirmesine, işlem verilerini doğrudan banka hesaplarından almasına, hesapların mutabakatını yapmasına ve gerçek zamanlı mali raporlar oluşturmasına olanak tanır. Açık Bankacılık, ticari bankacılığı ve finansal yönetimi kolaylaştırır.

Finansal hizmetler: Açık bankacılık, yeni finansal hizmetler geliştirmek için kullanılabilir. Örneğin, üçüncü taraf kuruluşlar, müşterilerin finansal verilerini analiz ederek onlara kişiselleştirilmiş tavsiyeler verebilirler.

Finansal teknoloji: Açık bankacılık, Fintek şirketlerinin yeni ürünler ve hizmetler geliştirmesini sağlar. Fintek şirketleri, açık bankacılık API 'lerini kullanarak müşterilerin finansal verilerine erişebilir ve bu verileri yenilikçi çözümler geliştirmek için kullanabilirler.



Şekil 2. Mobil Uygulamada Açık Bankacılık Platformu (Sipay, 2024)

2.1.Platformun Oluşturduğu Faydalar

Açık bankacılık hem müşteriler hem de işletmeler için fayda sağlar.

Müşteriler için açık bankacılığın sağladığı faydalar:

Daha iyi kişisel finans yönetimi: Açık bankacılık, müşterilerin finansal verilerini tek bir platformdan görüntüleyerek harcamalarını takip etmelerine, yatırımlarını yönetmelerine ve bütçe yapmalarına yardımcı olur.

Daha fazla seçenek: Açık bankacılık, müşterilerin daha fazla finansal hizmet seçeneğinden faydalanmalarına olanak tanır. Örneğin, müşteriler, kendi bankaları dışındaki hesaplardan para transferi yapabilir veya kişiselleştirilmiş tavsiyeler alabilirler.

Daha uygun maliyet: Açık bankacılık, finansal hizmetler sektöründe rekabeti artırmaya yardımcı olur. Bu da müşterilerin daha iyi hizmet ve daha düşük fiyatlardan yararlanmalarına yol açabilir.

İşletmeler için açık bankacılığın sağladığı faydalar:

Daha yüksek müşteri deneyimi: Açık bankacılık, işletmelerin müşterilerine daha iyi bir müşteri deneyimi sunmasına yardımcı olur. Örneğin, işletmeler, müşterilerin finansal verilerine erişerek onlara kişiselleştirilmiş teklifler sunabilirler.

Yeni iş fırsatları: Açık bankacılık, işletmeler için yeni iş fırsatları yaratır. Örneğin, işletmeler, açık bankacılık API'lerini kullanarak yeni ürünler ve hizmetler geliştirebilirler.

Daha fazla verimlilik: Açık bankacılık, işletmelerin daha verimli çalışmasını sağlar. Örneğin, işletmeler, açık bankacılık API'lerini kullanarak manuel işlemleri otomatikleştirebilirler.

Açık bankacılık, hızla gelişen bir alan ve gelecekte daha da yaygınlaşacağı öngörülmektedir.

2.2. Açık Bankacılık Ürün Geliştirme Sürecinde Yenilikçi Yaklaşımlar

Ar-Ge, yeni bilgi ve teknolojiler yaratmak, mevcut ürün, hizmet veya süreçleri iyileştirmek amacıyla gerçekleştirilen bilimsel ve teknik faaliyetlerdir. Bu çalışmalar, inovasyon üretimi ve verimliliği artırmak için yapılan sistematik araştırma ve geliştirme süreçlerini içerir (Tidd & Bessant, 2020). Tasarlanan açık bankacılık ürününün yenilikçi yönleri aşağıda sunulmaktadır:

Teknolojik yenilik: Açık bankacılık, yeni teknolojilerin kullanılmasını gerektiren bir sistemdir. Bu teknolojiler, yapay zekâ, büyük veri analizi ve blok zinciri gibi teknolojilerdir. Bu

teknolojiler, açık bankacılık ürünlerinin geliştirilmesinde ve uygulanmasında önemli bir rol oynamaktadır.

Müşteri odaklılık: Açık bankacılık ürünleri, müşterilerin ihtiyaçlarına ve beklentilerine yönelik olarak geliştirilmelidir. Bu ürünler, müşterilerin finansal verilerini yönetmelerine, finansal karar vermelerine ve yeni finansal hizmetlerden yararlanmalarına yardımcı olmalıdır.

Güvenlik: Açık bankacılık ürünleri, yüksek güvenlik standartlarına sahip olmalıdır. Bu ürünler, müşterilerin finansal verilerinin güvenliğini sağlamak için gerekli önlemleri almalıdır.

Bu nitelikler, açık bankacılık ürünlerinin başarılı bir şekilde geliştirilmesini ve uygulanmasını sağlayacaktır. Açık bankacılık, hızla gelişen bir alandır. Bu alandaki gelişmeler, yeni finansal hizmetler ve ürünlerin geliştirilmesini sağlayacaktır.

Şekil 3. Açık Bankacılık Ekosistemi (UK Open Banking Group, 2017)

2.3. Güvenlik

Güven, bankacılık sektörünün temel unsurlarından biridir ve sıkı kurallarla düzenlenmiştir. Müşteriler, paralarını bankalara emanet ederken bu güven duygusunu hissetmek istemektedirler. Bu

nedenle, güvenilirlik sağlamak, bankaların en önemli görevlerinden biri olarak öne çıkmaktadır (Özen, 2014).

Dijitalleşmenin hız kazanması, güvenlik alanındaki teknolojik ilerlemeler ve banka müşterilerinin artan talepleri, banka hizmetlerine çeşitli platformlar üzerinden erişimin zorunlu hale gelmesine neden olmuştur. Bu durum, finansal hizmet sağlayıcılarının sundukları çözümleri çeşitlendirerek, dijital kanalları daha etkili bir şekilde kullanmalarını gerektirir. Müşteriler artık bankacılık hizmetlerine yalnızca fiziksel şubeler aracılığıyla değil, aynı zamanda mobil uygulamalar ve web platformları gibi farklı dijital kanallardan ulaşmayı beklemektedirler (Haksever & Baykal, 2023).

Bu dönüşüm, bankaların daha esnek ve yenilikçi hizmetler sunmasını teşvik ederken, müşteri memnuniyetini artırma açısından da kritik bir adım oluşturmaktadır. Dijital bankacılık ve fintek uygulamaları, geleneksel bankacılık hizmetlerinin ötesine geçerek, kullanıcı deneyimini iyileştiren ve güvenliği güçlendiren çözümler sunma kapasitesine sahiptir.

Veri Güvenliği ve Gizlilik Stratejileri: Müşteri gizliliğinin korunması, açık bankacılık platformunun sürdürülebilirliği açısından da temel bir gerekliliktir. Kullanıcıların finansal verilerini güvenli bir şekilde üçüncü taraflarla paylaşmasını sağlayan bazı stratejiler geliştirilir. Bu stratejiler, kullanıcı güvenini korur, siber saldırılara karşı önlem alır ve regülasyonlara uyum sağlamayı zorunlu kılar. Bu stratejiler; güçlü şifreleme teknikleri (TLS(Transport Layer Security=Taşıma Katmanı Güvenliği)), AES(Advanced Encryption Standard=Gelişmiş Şifreleme Standardı)), güvenli API'ler ve kimlik doğrulama yöntemleri (OAuth 2.0(Open Authorization 2.0, çok faktörlü kimlik doğrulama)

kullanılarak sağlanır. Sıfır güven mimarisi benimsenir; bu, her erişim talebinin doğrulanmasını ve sürekli izlenmesini içerir. Ayrıca, PSD2(Payment Services Directive 2 = 2. Ödeme Hizmetleri Direktifi) ve GDPR(General Data Protection Regulation = Genel Veri Koruma Yönetmeliği) gibi düzenlemelere uyum, veri gizliliği ve güvenliğini artırır (Help Net Security, 2022).

Müşteri Güveni: Açık bankacılığın benimsenmesi, müşteri güveninin artırılmasıyla doğrudan ilişkilidir. Müşterilerin, finansal verilerinin üçüncü taraflarla paylaşımında güvence arayışı, bankaların ve fintek şirketlerinin güvenlik önlemlerini artırmasını zorunlu kılmaktadır.

API Güvenliği: API'lerin güvenliğini sağlamak için önerilen en iyi uygulamalar arasında, sağlam şifreleme yöntemleri, çok faktörlü kimlik doğrulama ve veri koruma yasalarına sıkı uyum yer almaktadır. Bu önlemler, müşteri verilerinin güvenli bir şekilde paylaşılmasını sağlamak için kritik öneme sahiptir.

Regülasyon ve Uyumluluk: Açık bankacılığın getirdiği regülasyon ve uyumluluk gereklilikleri, finansal kurumların veri koruma yasalarına uyum sağlamasını ve müşteri rızası gereksinimlerini yerine getirmesini zorunlu kılmaktadır.

3.Sonuç

Bu çalışma, dijitalleşmede hızını artırdığı finans sektöründe yenilikçi bir çözüm olan açık bankacılığın hem bireysel hem de kurumsal müşteriler için sunduğu potansiyel yararları ele almıştır. Açık bankacılık, finansal verilerin müşterilerin kontrolü altında olması ve bu verilerin üçüncü taraflarla güvenli bir şekilde paylaşılmasına olanak tanıyan API tabanlı bir sistemdir. Bu yaklaşım, bankaların ve Fintek şirketlerinin yenilikçi ürünler ve

hizmetler geliştirmesine imkân tanırken, finansal ekosistemi daha rekabetçi, şeffaf ve müşteri odaklı bir hale getirmektedir.

Araştırma kapsamında, Türkiye'nin genç ve teknolojiye yatkın nüfus yapısı ile büyüyen ekonomisinin açık bankacılığın gelişimi için elverişli bir zemin sunduğu vurgulanmıştır. Bankacılık ve Fintek sektörlerinde yaşanan dijital dönüşümün, müşteri deneyimini iyileştirme, operasyonel verimliliği artırma ve yeni iş modelleri geliştirme konularında önemli fırsatlar sunduğu görülmüştür.

Açık bankacılık platformlarının kullanıcıların tüm finansal işlemlerini tek bir platformda birleştirerek daha iyi bir kişisel finans yönetimine olanak tanıması, gerçek zamanlı ödemeler ve kredi değerlendirmelerinde yenilikçi yaklaşımları mümkün kılması, bu sistemin gelecekte finansal hizmetlerde temel bir yapı taşı olacağını göstermektedir. Aynı zamanda bu platformların müşteri gizliliği ve veri güvenliği konularındaki hassasiyetleri, bu sistemlerin güvenilirliğini artırmaktadır.

Bu bağlamda, açık bankacılık platformlarının geliştirilmesinde önerilen modüler ve ölçeklenebilir mimari yapılar, gelecekte uluslararası finansal sistemlerle entegrasyonun kolaylaşmasını sağlayacak şekilde tasarlanmalıdır. Güçlü API standartlarının benimsenmesi ve şifreleme teknolojilerinin iyileştirilmesi, veri güvenliği ve müşteri gizliliği konularında sektöre rehberlik edecek stratejik çözümler arasında yer almaktadır.

Sonuç olarak, açık bankacılık hem bireyler hem de kurumlar için finansal süreçleri kolaylaştıran ve verimliliği artıran bir yenilik olarak öne çıkmaktadır. Türkiye’nin bankacılık ve Fintek sektörlerindeki güçlü altyapısı ve dijitalleşme konusundaki yüksek potansiyeli, açık bankacılık uygulamalarının geliştirilmesi ve benimsenmesi için stratejik fırsatlar sunmaktadır. Bu alandaki

yenilikçi çalışmaların hem yerel hem de küresel düzeyde finansal hizmetlerin dönüşümüne katkı sağlayacağı açıktır.

KAYNAKÇA

Bilgel, D., & Aksoy, B. (2019). Finansal Teknoloji Şirketleri ve Geleceğin Bankacılığı: Açık Bankacılık. Avrupa Bilim ve Teknoloji Dergisi.

Deloitte. (2017). How to Flourish in an Uncertain Future: Open Banking and PSD2.

Demirez, D., Gür, D., & Yaşa Özeltürkay, E. (2021). BANKACILIK SEKTÖRÜNDE DJİTAL DÖNÜŞÜM: AÇIK BANKACILIK VE UYGULAMALARINA İLİŞKİN KAVRAMSAL BİR ARAŞTIRMA. Tarsus Üniversitesi Uygulamalı Bilimler Fakültesi Dergisi.

Fidanboy, Ö. (2022). Dijitalleşme süreci ve açık bankacılık.

Fintech Istanbul. (2017). *Açık bankacılık hakkında sorular ve cevapları*. https://fintechistanbul.org/2018/01/23/acik-bankacilik-hakkinda-sorular-ve-cevaplari/ adresinden alındı

Haksever, B., & Baykal, B. (2023). Bankacılık Sektöründe Dijitalleşme ve Finansal Teknolojilerin Hizmet Pazarlamasına Etkileri. Akademik Açı.

Help Net Security. (2022). *Open banking API security: Best practices to ensure a safe journey*. https://www.helpnetsecurity.com/2022/10/20/open-banking-api-security-best-practices/ adresinden alındı

iyzico, Dogma Alares, & ETİD. (2024). 2023 Türkiye e-ticaret ekosistemi.

Köseoğlu, Z. (2023). Açık Bankacılık Uygulamasının Türkiye Bankacılık Sektörüne Olası Etkileri. İstanbul.

McKinsey&Company. (2017, July). Data sharing and open banking. McKinsey&Company.

Özen, E. (2014). Bireysel Bankacılıkta Güven, Önem ve Maliyet Algılarının Sadakat Algısına Etkisi. Finans Politik & Ekonomik Yorumlar.

Saunders, A., & Cornett, M. (2019). *Financial Institutions Management: A Risk Management Approach.* Mc Graw Hill Education.

Schueffel, P. (2016). Taming the Beast: A Scientific Definition of Fintech. Switzerland: Journal of Innovation Management.

Sezal, L. (2021). Dünyada ve Türkiye'de Açık Bankacılık Uygulamaları Ve Yasal Düzenlemeler. Journal of Social and Humanities Sciences Research.

Sipay. (2024). https://sipay.com.tr/yardim-merkezi/acik-bankacilik/ adresinden alındı

T.C. Cumhurbaşkanlığı Finans Ofisi. (2024). *2023 Türkiye Fintek Ekosistemi Durum Raporu.*

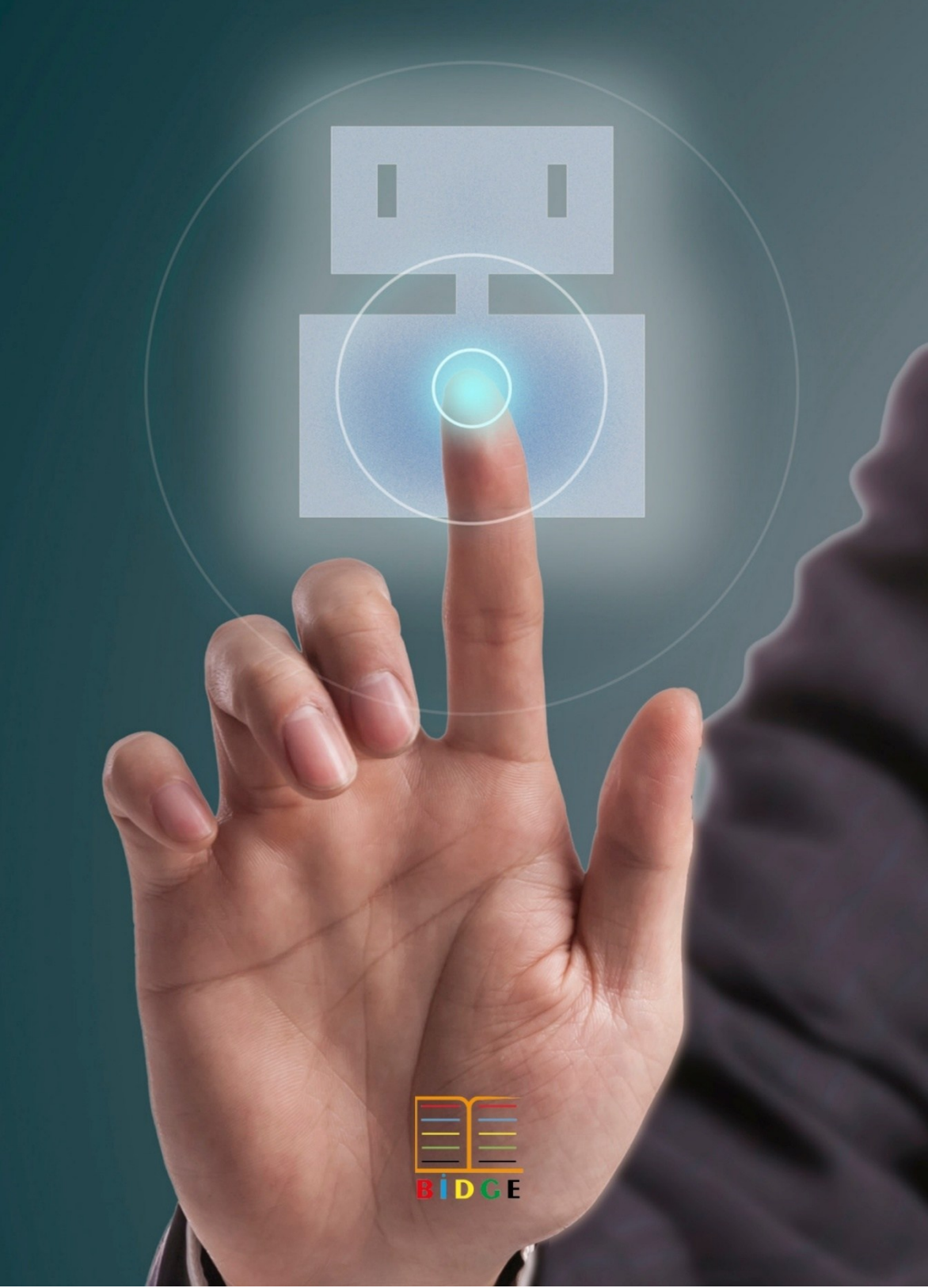
Tidd, J., & Bessant, J. (2020). *Managing innovation: integrating technological, market and organizational change.*

Türkiye Bankalar Birliği. (2022, Eylül). Dijital, İnternet ve Mobil Bankacılık İstatistikleri. Türkiye Bankalar Birliği.

Türkiye Bankalar Birliği. (2024). *Banka, Çalışan ve Şube Sayıları.*

UK Open Banking Group. (2017). *Open Banking Standards.* https://standards.openbanking.org.uk/ adresinden alındı

Uşaklı, B. (2021, Nisan). Finansal Hizmetler İnovasyonu: Açık Bankacılık.


BİDGE