

PRATİK YAPAY ZEKA UYGULAMALAR - I

Editör
Eyyüp Gülbandılar

BİDGE Yayınları

Pratik Yapay Zeka Uygulamaları 1

Editör: Prof.Dr.Eyyüp GÜLBANDILAR

ISBN: 978-625-372-395-8

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

Makine Öğrenmesi ve Derin Öğrenme Algoritmalarının Kredi Kartı Dolandırıcılığı Tespitindeki Başarılarının İncelenmesi	5
Dilek GÜN	5
Elif ÖZCEYLAN	5
Ayşe Berna ALTINEL	5
Kanser Genomisinde Yapay Zeka ve Makine Öğrenimi	28
Salih ACAR	28
Gıyasettin ÖZCAN	28
Eyyüp GÜLBANDILAR	28
Türkçe Metinlerde Anahtar Kelime Çıkarımı: LDA, RAKE ve BERTurk Yöntemlerinin Karşılaştırmalı Analizi	55
Kadir TOHMA	55
Halil İbrahim OKUR	55
Türk Dilleri İçin Doğal Dil İşleme: Zorluklar, Ortak Özellikler ve Çözümler	75
Halil İbrahim OKUR	75
Kadir TOHMA	75
Farksal Gelişim (DE) Algoritması	98
Uğur TALAŞ	98
Derin Öğrenme Teknikleri ve Uygulamaları	121
Alper Talha KARADENİZ	121
Güncel metasezgisel optimizasyon algoritmalarının CEC2020 test fonksiyonları ile değerlendirilmesi	153
Rabia UYAR	153

Mobil Uygulama İle Domates Yaprak Görüntülerinden Hastalık Teşhisi	182
Oğuzhan SEÇGEL	182
Kamil ORMAN	182
Spor Kulüpleri İçin Hizmet Olarak Beyaz Etiketli Platform	199
Ozan ÜST	199
Robbert OOMENS	199
Akıllı Şehirler: Bibliyometrik Analiz	206
Nevin AYDIN	206

BÖLÜM I

Makine Öğrenmesi ve Derin Öğrenme Algoritmalarının Kredi Kartı Dolandırıcılığı Tespitindeki Başarılarının İncelenmesi

Dilek GÜN¹
Elif ÖZCEYLAN²
Ayşe Berna ALTINEL³

1.Giriş

Kredi kartı, kullanım kolaylığı ve taksitlendirme olanakları nedeniyle çevrimiçi işlemlerde sıklıkla tercih edilen bir ödeme yöntemidir. Ancak, kredi kartlarının artan popülaritesi, beraberinde kötüye kullanım potansiyelini de getirmektedir. Dolandırıcılık

¹ Piri Reis Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul/Türkiye, Orcid: 0000-0002-4299-5363, dgun@pirireis.edu.tr

² İstanbul Arel Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul/Türkiye, Orcid: 0000-0003-3216-1979, elifozceylan@arel.edu.tr

³ Marmara Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul/Türkiye, Orcid: 0000-0001-5544-0925, berna.altinel@marmara.edu.tr

faaliyetlerindeki artış, bu duruma karşı alınması gereken önlemleri de zorunlu kılmaktadır.

Kredi kartı dolandırıcılığı, kart sahibinin bilgisi dışında birinin, kartın çalınması, kaybolması veya sahte bir kart üretilmesi gibi yollarla kart veya hesap bilgilerini kullanarak yasa dışı işlemler yaptığı bir tür kimlik hırsızlığı olarak tanımlanabilir.

Dolandırıcılık tespiti, özünde işlemleri meşru veya hileli olmak üzere doğru bir şekilde iki sınıfa ayırmayı amaçlayan bir veri madenciliği problemidir (Duman & Ozcelik, 2011).

Kredi kartı dolandırıcılığının iki ana türü vardır. Bunlardan ilki organize suç gruplarının gerçekleştirdiği sahte dolandırıcılıktır. Bunların toplam etkisi çok büyüktür ve genellikle bir bankanın onlarca, hatta yüzlerce müşterisini aynı anda etkilemektedir. İkinci tip kredi kartı dolandırıcılığı, kaybolan veya çalınan bir kartın yasa dışı kullanımıdır. Bu tür dolandırıcılıklar çoğunlukla suç grupları ile ilgili olmayıp, her dolandırıcılık faaliyeti bir veya birkaç kişiyi etkilemektedir (Duman & Ozcelik, 2011).

Sahte kredi kartı ve banka kartı işlemleri gibi sahtekarlık içeren finansal işlemler, kart veren bankalara hem finansal kayıplar hem de itibar kaybı nedeniyle ciddi zararlar verdiği için çeşitli araştırma topluluklarının geniş kapsamlı dikkatini çekmektedir. Kartları kullanarak sahtekarlık işlemlerini önlemek ve tespit etmek için birçok teknoloji önerilmiş olmasına rağmen, finansal işlem olanaklarının yaygınlığı ve evrensel doğası nedeniyle sahte işlemler hala yaygındır (Heryadi & Warnars, 2017).

Dolandırıcılığın tespiti, karmaşık bir hesaplama görevi olup, herhangi bir işlemi kesin olarak dolandırıcılık olarak tahmin eden bir

sistem hala mevcut değildir. Mevcut sistemler sadece bir işlemin dolandırıcılık olma olasılığını tahmin ederler. İyi bir dolandırıcılık tespit sisteminin özellikleri ise şu şekildedir (Zareapoor, Seeja, & Alam, 2012):

- Dolandırıcılıkları doğru bir şekilde tanımlamalıdır.
- Dolandırıcılıkları hızlı bir şekilde tespit etmelidir.
- Gerçek bir işlemi dolandırıcılık olarak sınıflandırmamalıdır.

Şirketler, müşteri profillemeye sinir ağı sistemlerinin etkinliğini ve karmaşıklığını sürekli olarak artırmaktadır. Bu sistemler, alışılmadık harcamaları ve potansiyel sahtekarlık içeren işlemleri erken aşamada tanımlayabilme kapasitesine sahiptir (B. Wiese & Omlin, 2009). Bu sayede, olası dolandırıcılık girişimleri hızlı bir şekilde tespit edilerek gerekli önlemler alınabilir ve müşteri güvenliği sağlanabilir.

Yaygın olarak kullanılan dolandırıcılık tespit yöntemleri arasında kural çıkarma teknikleri, karar ağaçları, Destek Vektör Makineleri (SVM), Lojistik Regresyon (LR), Yapay Sinir Ağları (ANNs) ve k-ortalamlar kümesi, genetik algoritmalar ve en yakın komşu algoritmaları gibi metasezgisel algoritmalar bulunmaktadır (Chaudhary, Yadav, & Mallick, 2012). Bu çalışmada gerçek bir veri seti kullanılarak SVM, NB, LSTM ve CNN olmak üzere dört farklı algoritma yardımıyla dolandırıcılık olan ve olmayan işlemlerin sınıflandırılması amaçlanmıştır.

Çalışmanın kalanı ise aşağıdaki şekilde düzenlenmiştir. Literatürde yapılan önceki çalışmalara 2.bölümde değinilmiş olup,

makalenin içeriğinde kullanılmış olan materyaller ve metotlara ise 3. bölümde değinilmiştir. Çalışmada gerçekleştirilen deneylerin sonuçları 4. bölümde paylaşılmış olup, son bölümde ise elde edilen sonuçların kısa bir özeti paylaşarak ileri ki çalışmalarda yapılması planlananlar tartışılmıştır.

2.Literatür Özeti

Dolandırıcılık, modern teknolojinin gelişmesi ve küresel iletişimin yaygınlaşmasıyla önemli ölçüde artmıştır, bu da her yıl dünya genelinde milyarlarca doların kaybolmasına neden olmaktadır. Dolandırıcılığı azaltmanın en iyi yolu önleme teknolojileri olsa da sahtekarlar uyum sağlayıcıdır ve zaman verildiğinde genellikle bu tür önlemleri atlatmanın yollarını bulacaklardır. Dolandırıcılığın tespit edilmesine yönelik metodolojiler, önleyici tedbirlerin başarısız olduğu durumlarda dolandırıcıları yakalamak önemlidir. İstatistik ve makine öğrenimi, sahtekarlık tespiti için etkili teknolojiler sağlar ve para aklama, e-ticaret kredi kartı dolandırıcılığı, telekomünikasyon dolandırıcılığı ve bilgisayar kırılması gibi faaliyetleri tespit etmek için başarıyla uygulanmaktadır (Bolton & Hand, 2002). Makine Öğrenimi modellerinin birçok alanda sınıflandırma problemlerinin çözümünde gösterdiği başarılı sonuçlar, birçok araştırmacıyı bu modelleri hileli işlemleri tanımak için kullanmaya yöneltmiştir (Heryadi & Warnars, 2017).

Bolton ve Hand (Bolton & Hand, 2002), genel olarak sahtekarlık sorununa ve sahtekarlık tespitinde karşılaşılan zorluklara dair iyi bir genel bakış sunmaktadır; özellikle kredi kartı

dolandırıcılığıyla ilgili bölümleri öne çıkmaktadır (B. J. Wiese, 2007).

2012 yılında yapılan bir çalışmada (Zareapoor et al., 2012); kredi kartı dolandırıcılığını tespit etmek için en yaygın kullanılan dokuz dolandırıcılık tespit yönteminin (karar ağacı, sinir ağı, Bayes ağı, genetik algoritma, destek vektör makinesi, k-en yakın komşu ve Yapay Bağışıklık Sistemi, gizli Markov modeli, bulanık sinir ağı ve bulanık Darwinist sistem) karşılaştırmalı bir çalışması sunulmuştur. İlgili çalışmada bahsi geçen makine öğrenmesi algoritmalarının birbiriyle karşılaştırılması verilmesine rağmen çalışmada gözden geçirilen makalelerde üretilen ölçülebilir performans ölçütleri hakkında bir tartışmanın olmaması bir eksiklik ve kapsanan tekniklerin sınıflandırma doğruluklarını kısaca tartışarak önceki çalışmalardaki boşlukları 2017 yılında yapılan bir çalışmada (Adewumi & Akinyelu, 2017) kısmen giderilmektedir.

Yapılan diğer bir çalışmada (Sahin & Duman, 2011), SVM tabanlı ve karar ağacı tabanlı kredi kartı dolandırıcılık tespit sistemleri arasında karşılaştırmalı bir çalışma gerçekleştirmiştir. Bu çalışmada kullanılan kredi kartı verileri, gerekli izinlerle bir ulusal bankanın kredi kartı veri depolarından alınmıştır ve ilgili çalışma kredi kartı dolandırıcılığı tespitinde SVM ve karar ağacı yöntemlerinin performansını gerçek veri seti ile karşılaştıran ilk çalışmalardan biridir. Bu çalışmada oluşturulan SVM modelleri polinomial, sigmoid, radyal tabanlı ve doğrusal çekirdek fonksiyonları olmak üzere dört farklı çekirdek (Kernel) fonksiyonu kullanılmaktadır. Uygulama sırasında, önce kullanılan veri kümesini üç gruba ayırmışlardır. Her bir grupta, veri kümesinin %70'i eğitim ve %30'u test için kullanılmıştır. Yazarlar, yedi farklı SVM tabanlı

ve karar ağacı tabanlı model geliřtirmiş ve her birini test etmişlerdir. Deneylerden elde edilen sonuçlar, karar ağacı tabanlı modelin SVM modelinden daha iyi performans gösterdiğini ortaya koymuştur. Modeller, SVM için %83.02 ile %93.08 arasında, Karar ağacı (DT) için ise %86.79 ile %94.76 arasında deęişen sınıflandırma doęruluęu üretmiştir.

Ayrıca, son derece dengesiz veriler sahtekarlık tespitinde başka bir sorundur. Egemen olan sınıf için rastgele az örnekleme yöntemi (undersampling), azınlık oranını ayarlamak için yaygın bir tekniktir. Ne yazık ki, bu yöntem kaçınılmaz olarak deęerli bilgilerin kaybolmasına neden olacaktır. Bir dięer makalede (Fu, Cheng, Tu, & Zhang, 2016), maliyet tabanlı bir örnekleme yöntemi kullanarak gerçek sahtekarlık vakalarından sentetik sahte sahtekarlık örnekleri üretilmiştir. Bu yöntem, eğitim için meşru işlemlerle karşılaştırılabilir sayıda sahtekarlık örneęi elde edilmesini sağlamaktadır. Ayrıca kredi kartı işlemlerindeki gizli dolandırıcılık desenlerini (pattern) arařtırmak için CNN tabanlı bir çerçeve önerilmiştir. Karakteristik uzayda maliyet tabanlı örnekleme yöntemini birleřtirerek, son derece dengesiz örnek setleri hafifletilmiş ve bu da sahtekarlık tespitinde üstün bir performans sağlamıştır. Daha karmaşık sahtekarlık desenlerini belirlemek için ticaret entropisi adı verilen yeni bir ticaret özellięi önerilmiştir. Ayrıca ticari bir bankadan alınan gerçek kredi kartı işlem verileri kullanılarak CNN algoritmasını Neural Network (NN), SVM ve RF gibi dięer makine öğrenmesi algoritmaları ile karşılařtırdılar. Farklı örnek setlerinde CNN F-score 0.33 ile en iyi sonucu vermiştir.

Bir dięer çalışmada (Heryadi & Warnars, 2017), dengesiz girdi veri setinden kısa vadeli ve uzun vadeli örüntüleri (pattern)

öğrenmek için üç derin öğrenme modeli CNN, SLSTM ve CNN-LSTM kullanmışlardır. Çalışmada kullanılan veriler, 2016-2017 dönemine ait Endonezya'daki bir bankadan gelen dengesiz kart işlemleridir ve bu işlemler dolandırıcılık veya dolandırıcılık değil olarak etiketlenmiştir. Öte yandan, performans ölçütü olarak AUC kullanıldığında CNN'nin (%73-%77) CNN-LSTM den (%70-%72) daha iyi performans gösterdiği ancak, doğruluk (accuracy) performans metriği kullanıldığında, CNN-LSTM nin eğitim doğruluğu aralığının (%78-%89) yığılmış LSTM den (%76-%89) ve CNN'den (%72-%81) daha iyi performans gösterdiği sonucuna varmışlardır.

Bhattacharyya vd. (Bhattacharyya, Jha, Tharakunnel, & Westland, 2011), kredi kartı dolandırıcılığını daha iyi tespit etme (ve böylece kontrol etme ve kovuşturma) girişiminin bir parçası olarak iki gelişmiş veri madenciliği yaklaşımını, destek vektör makineleri ve rastgele ormanları, iyi bilinen lojistik regresyon ile değerlendirmektedir. Çalışma, uluslararası bir kredi kartı operasyonundan elde edilen gerçek işlem verilerine dayanmaktadır. İlgili deney sonuçları, yazarların RF nin genel olarak daha iyi performans gösterdiğini ortaya koymuştur ancak, SVM, hassasiyet ve F-skoru ölçütlerinde lojistik regresyonu geride bırakmıştır; SVM'nin doğruluğu (accuracy) %93.8 olup, lojistik regresyonun %94.7 ve RF'nin %96.2 doğruluğunun gerisinde kalmıştır. SVM, geri çağırma (recall) oranı açısından üç teknik arasında en düşük değere sahip olup, %52.4 iken RF'nin geri çağırma oranı %72.7'dir.

Diğer bir çalışmada (Sundarkumar & Ravi, 2015), veri dengesizliği problemini çözmek için k Ters En Yakın Komşu ve Tek Sınıflı Destek Vektör Makinesi (OCSVM) kullanarak yeni bir hibrit

yaklaşım önerilmiştir. Önerilen modelin etkinliğini göstermek için Otomobil Sigortası Dolandırıcılığı tespit veri seti ve müşteri Kredi Kartı Terk tahmin veri seti kullanılmıştır. Ayrıca makalede Karar Ağacı (DT), Destek Vektör Makinesi (SVM), Lojistik Regresyon (LR), Olasılıksal Sinir Ağı (PNN), Grup Yöntemi ile Veri İşleme (GMDH), Çok Katmanlı Algılayıcı (MLP) kullanarak 10 kat çapraz doğrulama yöntemi takip edilmiştir GMDH, SVM ve LR nin istatistiksel olarak farklı olmadığı ve GMDH nin ROC da çok yüksek Alan Altında Eğri (AUC) verdiği ortaya çıkmıştır. Sonuçları aynı veri seti üzerinde yapılmış diğer çalışmalarla kıyaslamışlardır. (Vasu & Ravi, 2011) ile karşılaştırıldığında, (Sundarkumar & Ravi, 2015) GMDH, SVM, LR ve DT nin duyarlılık (sensitivity) sırasıyla %83.1, %87.7, %83.45 ve %91.2 olup AUC açısından önemli sonuçlar verdiği gözlemlenmiştir. Bu sınıflandırıcıların AUC sonuçlarına baktığımızda, GMDH nin diğer sınıflandırıcılar arasında üstün sonuçlar verdiğini görülmüştür. (Vasu & Ravi, 2011) ile karşılaştırıldığında, 12 kural ile %91 duyarlılık elde etmişlerdir. (Farquad, Ravi, & Raju, 2012) ile karşılaştırıldığında, önerilen metot önemli sonuçlar vermiştir.

Çalışmada (Jurgovsky et al., 2018), yeni gelen işlemlerde dolandırıcılık tespit doğruluğunu artırmak amacıyla kredi kartı sahiplerinin geçmiş satın alma davranışlarını bir araya getirmek için bir araç olarak uzun kısa süreli bellek (LSTM) ağlarını kullanılmıştır. Makalede LSTM ve RF metotları karşılaştırılmıştır. Çalışma, çevrimdışı ve çevrimiçi işlemlerin ardışık işlemlerin sıralı karakterine göre çok farklı özellikler sergilediğini göstermiş ve çevrimdışı işlemler için, LSTM gizli ardışık desenleri başarılı bir şekilde modelleyerek sahtekarlık tespitini iyileştirmiştir. İşlem

geçmişini LSTM ile modellemek, tüm özellik setleri boyunca RF ile tespit edilen sahtekarlıklardan sürekli olarak farklı bir doğru pozitif kümesi ortaya koymuş, bunun da birleşik bir yaklaşımı gerektirdiği sonucuna varılmıştır.

Tablo 1: Litaratür Özeti Karşılaştırması

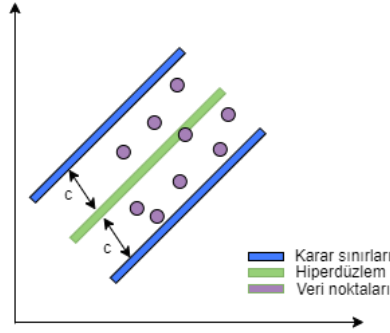
Çalışma	Algoritma	Performans
(Heryadi & Warnars, 2017)	CNN-LSTM	%89 (Accuracy), %77 (AUC)
(Fu et al., 2016)	CNN	%33 (F1-Score)
(Sundarkumar & Ravi, 2015)	SVM	%75.1 (Accuracy), %77.69 (AUC)
Farquad et al., 2012)	SVM	%74.79 (Accuracy), %78.55 (AUC)
(Sahin & Duman, 2011)	SVM	%83.02 ile %93.08 (Accuracy)
(Sahin & Duman, 2011)	DT	%86.79 ile %94.76 (Accuracy)
(Bhattacharyya et al., 2011)	SVM	%93.8(Accuracy), %52.4 (Recall)
(Vasu & Ravi, 2011)	SVM	%73.92 (Accuracy), %80.99 (AUC)

3. Materyal & Metot

3.1. Support Vector Machine (SVM)

1992 yılında, Boser ve diğerleri (Boser, Guyon, & Vapnik, 1992), optimal marj sınıflandırıcıları için bir eğitim algoritması önerdiler ve eğitim örnekleri ile sınıf sınırı arasındaki marjın maksimize edilmesinin, sınıflandırıcının genelleme performansı ile ilgili olarak maksimum kaybın minimize edilmesi anlamına geldiğini gösterdiler. Bu fikir başlangıçta ikili sınıf optimal marj sınıflandırıcılarının, ayırımın mümkün olduğu durumlarda, eğitim

verilerinin hatasız ayrımını sağladığı ve böyle bir sınıflandırıcıda aykırı değerlerin kolayca tanımlandığı için araştırılmıştır. Bu tür algoritmalara yönelik ilk araştırmalar ayrılabilir veri setlerine dayalıydı, ancak 1995 yılında Cortes ve Vapnik (Cortes & Vapnik, 1995), algoritmayı doğrusal olarak ayıramayan verileri dikkate alacak şekilde genişlettiler; kısa sürede bu sonuçların çok sınıflı sınıflandırma problemlerine de uygulanması için girişimlerde bulunuldu. Bu öğrenme makinesi daha sonra destek vektör makinesi (SVM) olarak adlandırıldı. SVM, güçlü ve sağlam bir teorik temele sahip bir makine öğrenme tekniğidir (B. Wiese & Omlin, 2009).



Şekil 1: SVM algoritması

3.2.Convolutional Neural Network (CNN)

CNN, ilk olarak LeCun ve diğerleri (LeCun, Bottou, Bengio, & Haffner, 1998) tarafından 1990 yıllarında tanıtılan ve sınıflandırma problemlerini çözmek için güçlü bir model haline gelen beslemeli bir sinir ağı türüdür. CNN özellik çıkarma bölümü ve sınıflandırma bölümü olmak üzere iki ana bölümden oluşur. Özellik çıkarma bölümü, bir veya birkaç evrişim katmanı ve havuzlama katmanından oluşur. Çıkarılan çıktı (özellik haritası),

sınıflandırma bölümünün girdisi haline gelir (Heryadi & Warnars, 2017).

CNN'nin sahte verileri tanımadaki yüksek performansı, önerilen işlem özelliklerinin, kısa vadeli ilişkilerle karakterize edilen finansal işlem paternlerini büyük ölçüde yakalayabildiğini göstermektedir. Kısa vadeli finansal işlemler, tipik sahtekarların çalıntı kartları kullanarak sahte işlemler yapmayı fazla beklemedikleri şeklinde yorumlanabilir (Heryadi & Warnars, 2017)

(3.1) eşitliğinde görüldüğü gibi; i CNN yapısındaki katman sayısını, K^i i . katmanın girişini, L^i i . katmanın çıkışını, $*$ evrişim operatörünü, W^i ağırlık matrisini, A^{i-1} önceki katmanın çıkışını, b^i bias'ı, h^i ise aktivasyon fonksiyonunu temsil etmektedir (Hong, Nguyen, Jung, & Ahn, 2021).

$$K^i = W^i * A^{i-1} + b^i ; L^i = h^i (K^i) \quad (3.1)$$

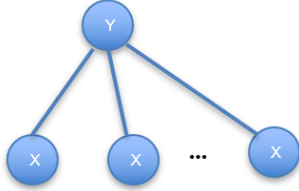
3.3.Naive Bayes (NB)

Naive Bayes, Bayes teoremini temel alan olasılık tabanlı bir makine öğrenmesi algoritmasıdır. (3.2) eşitliğine bakıldığında; $P(A)$ ve $P(B)$ sırasıyla A ve B olaylarının marjinal olasılıklarını, $P(A \setminus B)$ B olayı gerçekleştiğinde A olayının meydana gelme olasılığını; $P(B/A)$ A olayı gerçekleştiği durumda B olayının meydana gelme olasılığını ifade etmektedir (Vembandasamy, Sasipriya, & Deepa, 2015).

$$P(A \setminus B) = \frac{P(B \setminus A)P(A)}{P(B)} \quad (3.2)$$

Şekil 2'de Naive Bayes algoritmasındaki girdi (x) ve çıktı (y) değişkenleri düğümler ile temsil edilmekte olup, bu düğümler

arasındaki çizgiler ise iki düğüm arasındaki ilişkiyi temsil etmektedir.



Şekil 2: Naive Bayes algoritması

3.4. Long Short-Term Memory (LSTM)

Uzun Kısa Süreli Bellek (LSTM) modeli ilk olarak yeni, verimli, gradyan tabanlı bir yöntem olarak (Hochreiter & Schmidhuber, 1997) tarafından tanıtılmıştır. LSTM, kaybolan hata probleminden kurtulmak için tasarlanmıştır (Hochreiter, Bengio, Frasconi, & Schmidhuber, 2001). LSTM modeli, geri yayılım hatası sırasında uzun süreli bağımlılıklar nedeniyle oluşan gradyan patlaması veya zayıflaması problemlerini çözmek için (Hochreiter & Schmidhuber, 1997) tarafından önerilen tekrarlayan sinir ağı (Recurrent Neural Network (RNN)) modelinin bir çeşididir (Heryadi & Warnars, 2017).

(3.3) eşitliğinde görüldüğü üzere h_t t. zamandaki gizli durumu, o_t çıkış kapısını ve C_t ise hücre durumunu ifade etmektedir (Smagulova & James, 2019)

$$h_t = o_t \odot \tanh(C_t) \quad (3.3)$$

3.5. CNN ve LSTM Hibriti (CNN-LSTM)

Son beş yılda yapılan birçok çalışma, birleşik CNN ve LSTM modelinin sırasıyla CNN ve LSTM den daha sağlam bir model

ürettiğini göstermiştir (Heryadi & Warnars, 2017). CNN-LSTM modeli, giriş verilerinin ilk olarak CNN tarafından işlenmesi şeklinde çalışır. CNN çıktısı daha sonra sınıflandırıcı olarak LSTM tarafından işlenir (Heryadi & Warnars, 2017). CNN-LSTM modelinin yüksek performansı, CNN'in kısa vadeli özellik ilişkilerini ve LSTM nin daha uzun vadeli zamansal özellik ilişkilerini yakalama yeteneğini birleştirmesinden kaynaklanmaktadır (Heryadi & Warnars, 2017).

4.Deney Ortamı, Sonuçlar ve Değerlendirme

4.1. Deney Ortamı

Bu çalışmada kullanılan veri seti, kredi kartı dolandırıcılığı tespiti için kullanılan bir veri seti olup, Kaggle sitesi üzerinden elde edilmiştir¹.

Veri seti, Eylül 2013'te Avrupalı kart sahipleri tarafından kredi kartlarıyla yapılan işlemleri içermektedir. Bu veri seti, 284.807 işlemden 492'sinin dolandırıcılık olduğu iki gün içinde gerçekleşen işlemleri sunmaktadır.

Veri kümesi oldukça dengesizdir, pozitif sınıf (dolandırıcılıklar) tüm işlemlerin %0172 sini oluşturmaktadır. Veri seti incelendiğinde, V1'den V28'e kadar toplam 28 adet özelliğin gizlilik sebebiyle açıklaması yapılmamıştır. V1, V2, ..., V28 PCA yöntemi ile elde edilen ana bileşenlerdir (öznitelik). Yalnızca PCA dönüşümünün sonucu olan sayısal girdi değişkenlerini içerir. 'Zaman' ve 'Miktar' özellikleri PCA ile dönüştürülmeyen iki özniteliktir. Çıktı özelliği (Class) ile toplamda 31 öznitelik bulunmaktadır. Burada, 'Zaman' özniteliği, her işlem ile veri kümesindeki ilk işlem arasında geçen saniyeyi içermektedir. Aynı

zamanda ‘Miktar’ özniteliği ise işlem tutarıdır, bu özellik maliyete duyarlı öğrenme için kullanılır. ‘Sınıf’ özniteliği çıktı değişkenidir ve dolandırıcılık durumunda 1, aksi takdirde 0 değerini almaktadır. Çalışmada veri seti çok fazla örneklem içerdiğinden bu veri setindeki örneklem sayısı 5598’e indirgenerek kullanılmıştır. Tablo 2’de bu veri setine ait özet bilgiler gösterilmektedir.

Tablo 2: Veri setine ait özellikler

Public/ Private	Özellik Sayısı	Sınıf Sayısı	Sınıflardaki Örnek Sayısı	
Public	31	2	0	4894
			1	704

Veri setinde ön işleme adımında 3291. satırda bulunan V1 özelliğine ait “-109.287.386.429.334” verisi ve 779. satırda bulunan “128.821.131.189.778” verileri bozuk veriler olduğu için bu verilere ait iki satır silinmiştir. Şekil 3’te veri seti dağılımı grafik olarak gösterilmektedir.



Şekil 3: Veri setine ait örneklem dağılımı

Çalışmada ilk olarak veri seti girdi ve çıktı olarak ikiye ayrılmıştır (x, y). Burada y değişkeni class özneliğini temsil etmektedir. x ve y değişkenleri kendi içlerinde train (%80) ve test (%20) olarak ikiye ayrılmıştır. Bu ayırım yapılırken train ve test verileri aralarındaki sınıf etiketi dengesinin sağlanması için, train_test_split() fonksiyonuna ait bir parametre olan stratify parametresi kullanılmıştır. Model her çalıştırıldığında aynı sonuçları elde etmek için aynı fonksiyon içinde random state parametresi kullanılmıştır.

Bu çalışmadaki sonuçlar python dili kullanılarak elde edilmiş olup, Pycharm geliştirme ortamı kullanılmıştır. SVM ve NB algoritmaları için Sckit-Learn, LSTM ve CNN algoritması için ise Keras ve Tensorflow kütüphaneleri kullanılmıştır. LSTM modeli için epoch değeri 100, batch size değeri 32, dropout değeri 0,2 olarak seçilmiş olup, modeli kurmak için 50 nöronlu iki katman kullanılmıştır. CNN modeli için ise epoch değeri 100, batch size değeri 32, dropout değeri 0,5 olarak seçilmiştir.

Eğitim süresi her cihazın donanımsal özelliklerine göre değişiklik göstermekte olup, bu çalışmada sonuçların elde edilme aşamasında Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz, 2401 Mhz, 2 çekirdekli 53 işlemci, 16 GB RAM ve AMD Radeon(TM) R5 M330 10 GB GPU belleği kapasitesi ile donatılmış bir bilgisayar kullanılmıştır.

4.2. Deney Sonuçları ve Değerlendirme

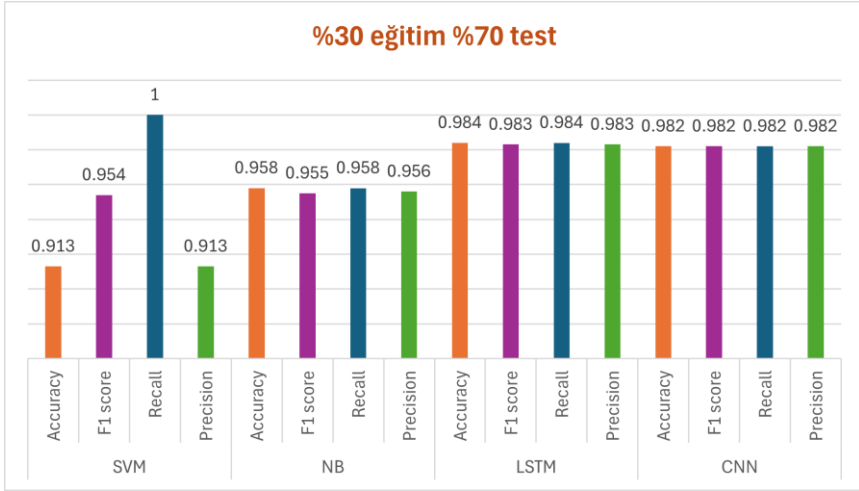
SVM, NB, LSTM ve CNN algoritmaları kullanılarak 4 farklı model elde edilmiştir.

Her bir algoritma kullanılarak model fit edilmiş ve x_{test} verilerine göre y değerleri tahmin edilmiştir. y çıktı verisi ile y 'nin gerçek değerleri arasında 4 farklı metrik (accuracy, recall, precision ve F1 score) kullanılarak, modelin performansı ölçülmüştür. Her bir metrik için average parametresi weighted olarak seçilmiştir. Elde edilen sonuçlar Tablo 3'te gösterilmektedir.

Tablo 3: Algoritma sonuçlarının karşılaştırılması

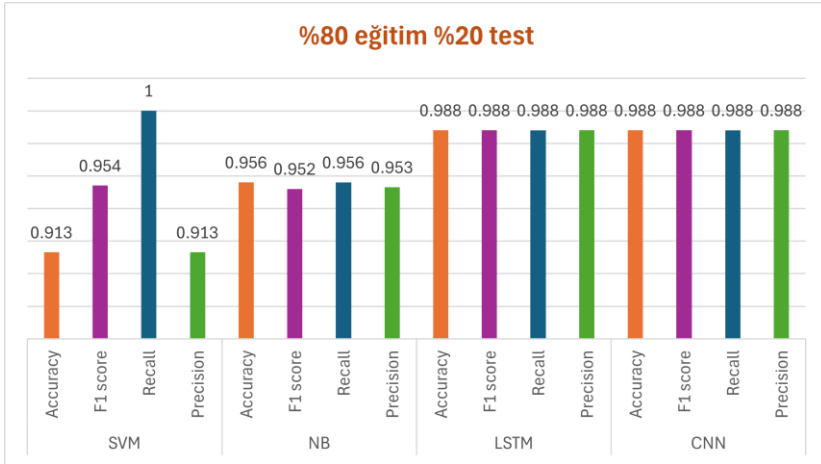
Algoritma	%30 eğitim %70 test	%40 eğitim %60 test	%80 eğitim %20 test
SVM	Accuracy: %91.3 F1 score: %95.4 Recall: %100 Precision: %91.3	Accuracy: %91.3 F1 score: %95.4 Recall: %100 Precision: %91.3	Accuracy: %91.3 F1 score: %95.4 Recall: %100 Precision: %91.3
NB	Accuracy: %95.8 F1 score: %95.5 Recall: %95.8 Precision: %95.6	Accuracy: %95.8 F1 score: %95.4 Recall: %95.8 Precision: %95.5	Accuracy: %95.6 F1 score: %95.2 Recall: %95.6 Precision: %95.3
LSTM	Accuracy: %98.4 F1 score: %98.3 Recall: %98.4 Precision: %98.3	Accuracy: %98.2 F1 score: %98.2 Recall: %98.2 Precision: %98.2	Accuracy: %98.8 F1 score: %98.8 Recall: %98.8 Precision: %98.8
CNN	Accuracy: %98.2 F1 score: %98.2 Recall: %98.2 Precision: %98.2	Accuracy: %98.2 F1 score: %98.2 Recall: %98.2 Precision: %98.2	Accuracy: %98.8 F1 score: %98.8 Recall: %98.8 Precision: %98.8

Tablo 3 ve Şekil 4'e bakıldığında, SVM algoritması için eğitim ve test oranlarının değiştirilmesi başarı yüzdelelerini etkilememiş olup, NB algoritması için en yüksek metrik sonuçları %30 eğitim ve %70 test oranlarında elde edilmiştir. Eğitim verisi oranı arttıkça tüm metriklerde kademeli bir şekilde başarı azalmasının olduğu görülmektedir.



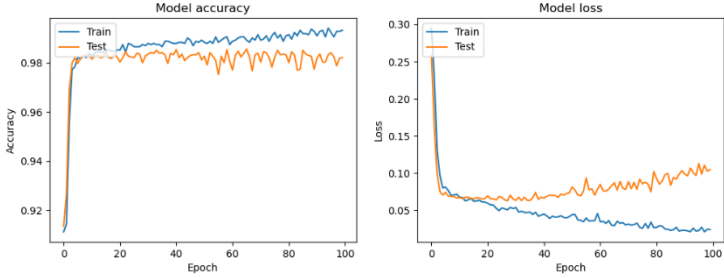
Şekil 4: %30 eğitim ve %70 test oranlarındaki deneysel sonuçlar

Tablo 3 ve Şekil 5'e bakıldığında ise LSTM ve CNN algoritmaları için en yüksek başarı oranları %80 eğitim %20 test oranlarında elde edilmiştir. Sonuçlar incelendiğinde bu iki algoritma için test verisi oranı arttıkça performansın arttığı görülmektedir.

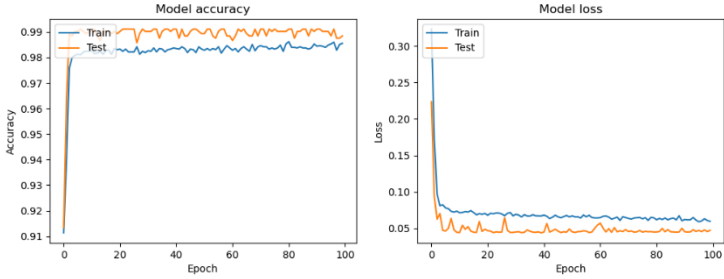


Şekil 5: %80 eğitim ve %20 test oranlarındaki deneysel sonuçlar

LSTM ve CNN algoritmalarının sonuçları karşılaştırıldığında ise %30 eğitim %70 test oranları hariç diğer veri seti ayrımlarında aynı sonuçların elde edildiği görülmektedir. %30 eğitim %70 test oranları kullanılarak elde edilen sonuçlara bakıldığında ise LSTM algoritmasının tüm metriklerde CNN algoritmasından daha yüksek başarı değerlerine sahip olduğu görülmektedir.



Şekil 6: CNN modeline ait accuracy ve loss değerleri grafiği



Şekil 7: LSTM modeline ait accuracy ve loss değerleri grafiği

Şekil 6 ve Şekil 7 incelendiğinde, elde edilen sonuçları destekler şekilde CNN modelinin LSTM modeline göre daha yüksek loss değerlerine sahip olduğu görülmektedir.

5.Tartışma ve Sonuç

Bu çalışmada kredi kartı dolandırıcılığı veri seti kullanılarak dolandırıcılık olan ve olmayan işlemlerin sınıflandırılması amaçlanmıştır. Bu amaç için SVM, NB, LSTM ve CNN olmak üzere 4 farklı algoritma kullanılmış olup, 4 farklı metrik üzerinde sonuçlar elde edilmiştir. Bunun için ilk olarak veri seti ön işlemde geçirilerek örneklem sayısı azaltılmış ve aykırı veriler temizlenmiştir. Ön işlemde geçirilen veri seti kullanılarak 3 farklı eğitim ve test veri oranları ile algoritmalar eğitilmiş ve performansları incelenmiştir. Elde edilen sonuçlar en yüksek başarıya sahip algoritmaların %80 eğitim ve %20 test oranlarında her metrikte 0.988 ile LSTM ve CNN modellerinden elde edildiğini göstermektedir. Eğitim ve test oranları bakımından karşılaştırıldığında ise NB algoritması için en yüksek başarı %30 eğitim %70 test oranlarında elde edilmiş, SVM algoritması için ise eğitim ve test oranlarının değişmesi başarıyı etkilememiştir. Sonuçlar incelendiğinde farklı oranlarda eğitim ve test verilerinin kullanılmasının performans üzerinde büyük etkilerinin olabileceği görülmektedir.

İleri ki çalışmalarda, hibrit olarak LSTM ve CNN algoritmalarının farklı eğitim ve test veri oranlarında kullanılarak kredi kartı dolandırıcılığı tespiti yapılması planlanmaktadır.

Referanslar

Adewumi, A. O., & Akinyelu, A. A. (2017). A survey of machine-learning and nature-inspired based credit card fraud detection techniques. *International Journal of System Assurance Engineering and Management*, 8, 937-953.

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision support systems*, 50(3), 602-613.

Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical science*, 17(3), 235-255.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). *A training algorithm for optimal margin classifiers*. Paper presented at the Proceedings of the fifth annual workshop on Computational learning theory.

Chaudhary, K., Yadav, J., & Mallick, B. (2012). A review of fraud detection techniques: Credit card. *International journal of computer applications*, 45(1), 39-44.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20, 273-297.

Duman, E., & Ozcelik, M. H. (2011). Detecting credit card fraud by genetic algorithm and scatter search. *Expert Systems with Applications*, 38(10), 13057-13063.

Farquad, M., Ravi, V., & Raju, S. B. (2012). Analytical CRM in banking and finance using SVM: a modified active learning-based rule extraction approach. *International Journal of Electronic Customer Relationship Management*, 6(1), 48-73.

Fu, K., Cheng, D., Tu, Y., & Zhang, L. (2016). *Credit card fraud detection using convolutional neural networks*. Paper presented at the Neural Information Processing: 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16–21, 2016, Proceedings, Part III 23.

Heryadi, Y., & Warnars, H. L. H. S. (2017). *Learning temporal representation of transaction amount for fraudulent transaction recognition using CNN, Stacked LSTM, and CNN-LSTM*. Paper presented at the 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom).

Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In: *A field guide to dynamical recurrent neural networks*. IEEE Press In.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.

Hong, S., Nguyen, H.-T., Jung, J., & Ahn, J. (2021). Seismic Ground Response Estimation Based on Convolutional Neural Networks (Cnn). *Applied Sciences*, 11(22), 10760.

Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.-E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. *Expert Systems with Applications*, 100, 234-245.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

Sahin, Y., & Duman, E. (2011). *Detecting credit card fraud by decision trees and support vector machines*. Paper presented at the Proceedings of the International MultiConference of Engineers and Computer Scientists.

Smagulova, K., & James, A. P. (2019). A survey on LSTM memristive neural network architectures and applications. *The European Physical Journal Special Topics*, 228(10), 2313-2324.

Sundarkumar, G. G., & Ravi, V. (2015). A novel hybrid undersampling method for mining unbalanced datasets in banking and insurance. *Engineering Applications of Artificial Intelligence*, 37, 368-377.

Vasu, M., & Ravi, V. (2011). A hybrid under-sampling approach for mining unbalanced datasets: applications to banking and insurance. *International Journal of Data Mining, Modelling and Management*, 3(1), 75-105.

Vembandasamy, K., Sasipriya, R., & Deepa, E. (2015). Heart diseases detection using Naive Bayes algorithm. *International Journal of Innovative Science, Engineering & Technology*, 2(9), 441-444.

Wiese, B., & Omlin, C. (2009). Credit card transactions, fraud detection, and machine learning: Modelling time with LSTM recurrent neural networks. In *Innovations in neural information paradigms and applications* (pp. 231-268): Springer.

Wiese, B. J. (2007). *Credit Card Transactions, Fraud Detection, and Machine Learning: Modelling Time with LSTM Recurrent Neural Networks*. (Master). University of the Western

Cape, Retrieved from
https://etd.uwc.ac.za/bitstream/handle/11394/9590/Wiese_M_NSC_2007.pdf?sequence=1&isAllowed=y%20tez

Zareapoor, M., Seeja, K., & Alam, M. A. (2012). Analysis on credit card fraud detection techniques: based on certain design criteria. *International journal of computer applications*, 52(3).

BÖLÜM II

Kanser Genomüğünde Yapay Zeka ve Makine Öğrenimi

Saliha ACAR¹
Gıyasettin ÖZCAN²
Eyyüp GÜLBANDILAR³

GİRİŞ

1950'li yılların başında Alan Turing'in “Makineler Düşünebilir mi?” sorusuyla makine öğrenmesi kavramı ortaya çıkmış ve yapay zeka terimi, 1956 yılında gerçekleştirilen Dartmouth Konferansında John McCarthy tarafından ilk kez kullanılmıştır (Grzybowski, Pawlikowska–Łagód & Lambert, 2024). Basit bir ifadeyle yapay

¹ Öğr.Gör., Bursa Teknik Üniversitesi, Bilgi İşlem Daire Bşk., Bursa/Türkiye, Orcid: 0000-0003-0297-4864, saliha.acar@btu.edu.tr

² Doç. Dr., Bursa Uludağ Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Bursa/Türkiye, Orcid: 0000-0002-1166-5919, gozcan@uludag.edu.tr

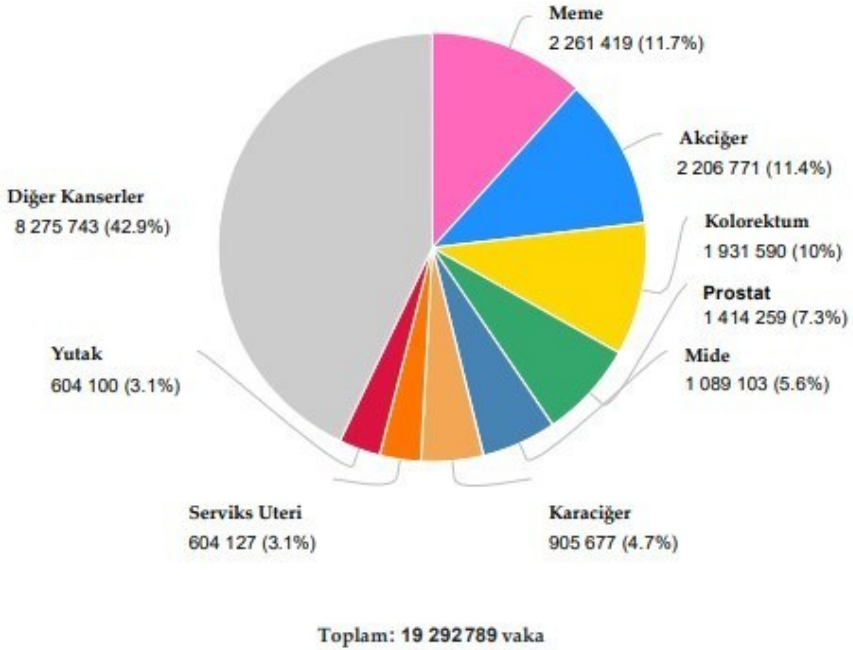
³ Prof. Dr., Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Eskişehir/Türkiye, Orcid: 0000-0001-5559-5281, egulbandilar@ogu.edu.tr

zeka, bir makinenin yeterli sayıda temsili örnekten kalıpları ve ilişkileri öğrenmesi ve bu bilgileri tanınması ile bu bilgiyi daha önce görmediği veriler üzerinde karar vermek için etkili bir şekilde kullanma yeteneği olarak tanımlanabilir. Yapay zeka, makine öğrenimi ve derin öğrenmeyi de kapsayan (ve bazen eşanlamlı olarak kullanılan) geniş bir terimdir. 1990'ların sonlarından itibaren popülerleşmeye başlayan makine öğrenimi ise yapay zekanın bir alt alanıdır ve tamamen bağlantılı birden fazla gizli katmana sahip yapay sinir ağlarına odaklanan derin öğrenme konseptini içine alır (Bhinder & ark., 2021).

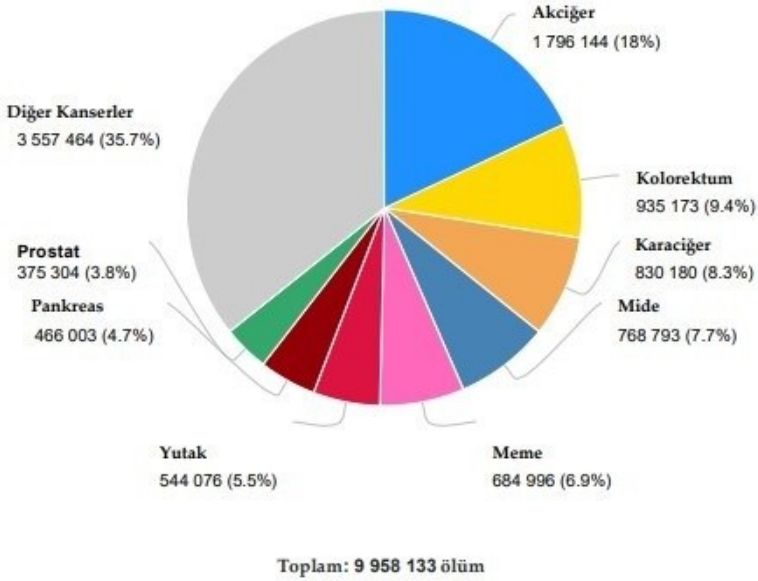
Açıkça programlanmaya ihtiyaç duymadan belirli bir görevi gerçekleştirebilmek, yapay zeka ve makine öğrenimi algoritmalarının ayırt edici özelliğidir (Caudai & ark., 2021). Bu özelliğinin yanı sıra yapay zeka ve veri madenciliğindeki sürekli iyileştirmeler ve gelişmeler, son birkaç on yılda biyomedikal araştırma alanlarında devrim niteliğinde değişikliklere neden olmuştur (Pathak & ark., 2023). Hastalıkların erken teşhisi, tedavi planlarının optimizasyonu, dijital ortamda ilaç keşfi ve tedavi yanıtlarının önceden tahmini bu değişikliklerden bazılarıdır.

Kardiyovasküler hastalıklardan sonra ikinci önde gelen ölüm nedeni olan kanserle ilgili genomik araştırmalarda da yapay zeka ve makine öğrenmesi yöntemleri etkin ve yaygın olarak kullanılmaktadır. GLOBOCAN 2020 verilerine göre yılda yaklaşık 19,3 milyon yeni vaka (Şekil 1) ve 10 milyon ölümün (Şekil 2) rapor edildiği bir hastalıklar bütünü olan kanserin genom düzeyinde araştırılmasını hızlandıran ve kolaylaştıran büyük boyutlu verilerin varlığı bu alandaki çalışmaların artmasını sağlamıştır (Arjmand & ark., 2022). Uzamsal ve tek hücre genomunun kullanılması,

tümörün daha kapsamlı anlaşılmasını kolaylaştırmak, insanlarda belirsiz patogenezi deşifre etmek ve bu mekanizmaya dayalı hedefe yönelik ilaçlar geliştirmek için yapılan çalışmalara önemli oranda katkıda bulunabilir. Yapay zeka ve genomik verilerin birleşimi kanser alt tiplerinin teşhisine, yeni biyobelirteçlerin ve ilaç hedeflerinin keşfedilmesine ve kansere neden olan genlerin daha iyi anlaşılmasına yardımcı olabilir (Liao & ark., 2023).



Şekil 1: GLOBOCAN 2020 verilerine göre dünyada kanser insidansları (Sung & ark., 2021)



Şekil 2: GLOBOCAN 2020 verilerine göre dünyada kanser mortaliteleri (Sung & ark., 2021)

KANSERE GENEL BİR BAKIŞ

Kanser biyolojisi, organizma seviyesinde adım adım ilerleyen oldukça karmaşık hücre içi, hücreler arası, dokular arası ve sistemler arası etkileşimlerden oluşur. Kanser belirli bir zaman diliminde, hücre genomunda artarda gelen mutasyonların (ya da epigenetik değişikliklerin) birikimi sonucunda ortaya çıkan bir patolojidir ve başlangıç (inisyasyon), gelişme (promosyon) ve ilerleme (progresyon) evrelerini içerir (Arjmand & ark., 2022).

Süreç boyunca kanser hücreleri çeşitli biyolojik özellikler kazanır. Bunlar;

- Genomik instabilite ve kromozomal anormallikler,

- Belirsiz büyüme ve kontrolsüz çoğalma potansiyeli,
- Replikatif ölümsüzlük,
- Büyüme baskılayıcılardan ve hücre ölümü sinyallerinden kaçınmak,
- Hücre dışı matrisi yeniden modellemek ve dinamik tümör mikroçevresi oluşturmak,
- Bağışıklık gözetimi ve yıkımından kaçınmak,
- Hücresel enerjileri yeniden programlamak,
- Tümörü teşvik eden inflamasyonu tetiklemek,
- Anjiyojenik anahtarın etkinleştirilmesi,
- İstila ve metastazın etkinleştirilmesi şeklinde sıralanabilir (Arjmand & ark., 2022).

Kanserin ilk kanıtlarından bazıları Eski Mısır'da fosilleşmiş kemik tümörleri arasında bulunmuştur. Kanser kelimesi Latince yengeç anlamına gelen “cancer” veya “carcinus” kelimelerinden türemiştir. MS 200 yılı civarında, Yunan doktor Galen ise, tüm tümör tipleri için Yunancada şişlik anlamına gelen “onkos” terimini kullanmıştır (Belciug, 2020).

İnsan vücudunu etkilediği bilinen 100'den fazla kanser türü bulunmaktadır. Birçok kanser türü başlangıçta belirti vermez ve kanserin tipine bağlı olarak görülen genel belirtiler değişiklik gösterebilir. Bu nedenle her kanser türüne yaklaşım farklı olmaktadır. Hem erkeklerde hem de kadınlarda en sık rastlanan ve ölüme neden olan kanser türü akciğer kanseridir. 2. sırada ise

erkeklerde prostat kanseri, kadınlarda ise meme kanseri gelmektedir (Baykara, 2016).

Kanserin oluşumunda en büyük role sahip olan 3 gen grubu bulunmaktadır. Bunlar hücre büyümesi ve ölümü gibi süreçleri düzenleyen onkogenler, hücrenin gardiyanları olarak isimlendirilen tümör baskılayıcı genler ve DNA tamir genleridir. RAS, BRAF ve MYC en iyi bilinen onkogenlerden; TP53, BRCA1 ve RB1 ise en iyi bilinen ve çalışılmış olan tümör baskılayıcı genlerdendir (Baykara, 2016).

Tümör baskılayıcı genlerin bazı işlevsizliklerinde dikkate alınan ilkelere birisi "çift vuruş" hipotezidir. Bu hipoteze göre onkogenlerden farklı olarak tümör baskılayıcı genler resesif olma eğilimindedir. Başka bir deyişle, bu genlerin alellerinden birisi inaktive olduğunda diğeri normal olan alel, yeterli ürün üreterek mutasyona uğramış alelin işlevsizliğini telafi edebilir. Bu nedenle kanser oluşumunda ve/veya gelişiminde rol oynaması için bu genlerin her iki alelinin de devre dışı kalması gerekir. Ancak tümör baskılayıcı genlerin inaktivasyonları genetik olmayan faktörlerden de kaynaklanabilmektedir (Arjmand & ark., 2022).

DNA tamir mekanizmaları da genomik kararlılığın devamını sağlayan sistemler olup bu sistemin bir parçası olan 100'den fazla gen tanımlanmıştır. ATM, ATRX, MLH1 ve MSH6 bunlardan bazılarıdır (Baykara, 2016).

KANSER GENOMİĞİ VE EPIGENOMİK

"Genom" kavramı ilk olarak 1920 yılında, o zamanlar Hamburg Üniversitesi Botanik profesörü olan Hans Winkler tarafından, çekirdekte bulunan "kromozomların haploid sayısına"

atıfta bulunularak önerilmiş, sonraları, dizilemedeki teknolojik ilerlemeler ve DNA karmaşıklığının keşfiyle birlikte bu kavram bir hücre veya organizmadaki tüm DNA dizilerini kapsayacak şekilde genişletilmiştir. Genom bilimi olarak tanımlanan “Genomik” terimi ise 1986 yılında Thomas Roderick tarafından türetilmiştir. İlk tam genom dizileme 1996 yılında yapılmış, insan genomunun dizilenmesi ise 13 yıl sürerek 2003 yılında tamamlanmıştır (Caudai & ark., 2021).

Epigenomik, epigenetik süreçleri genom düzeyinde inceleyen bir disiplindir. Bu süreçler DNA dizisindeki değişikliklerden bağımsız olarak gen aktivitesi ve kalıtımın düzenleyici mekanizmalarını içerir. Bunlar DNA metilasyonu, histon modifikasyonu, kromatin modifikasyonları ve kodlamayan RNA’ların işlevleri olarak tanımlanmaktadır (Caudai & ark., 2021). Epigenetik terimi 1942’de İngiliz Biyolog Conrad Waddington tarafından türetilmiştir. Epigenetik değişiklikler genlerin ifade seviyelerini değiştirerek bir genin aktivasyonuna ya da sessizleşmesine neden olabilirler.

Genomik ve epigenomik, klinik çalışmalar alanında özellikle önleme, tedavi ve takip uygulamalarında giderek daha önemli hale gelmiştir. Kanser genomiki; tümörlerin genetik, metabolik, sistemik ve çevresel düzeyde oluşumu ve yayılmasının altında yatan dinamikleri anlamak amacıyla, tümör hücreleri ve normal hücreler arasındaki DNA dizileri ve gen ekspresyonlarındaki farklılıkları incelemektedir. Büyük miktarlarda genomik, proteomik ve epigenomik bilginin varlığı ve bunların entegrasyonu kanser oluşumu gibi karmaşık dinamiklerin giderek daha kapsamlı bir şekilde araştırılmasına olanak sağlamıştır (Caudai & ark., 2021). Bu

amaca yönelik olarak yapay zeka modellerinde kullanılabilir olan kanser genomiği veritabanı kaynaklarının bir listesi Tablo 1’de verilmiştir.

Tablo 1. Yapay zeka modellerinde kullanılabilir kanser genomiği veritabanı kaynakları (Liao & ark., 2023)

Veritabanı Adı	Temel Özellikler	Web Erişim Linki
TCGA	Kapsamlı vt., büyük veri	https://www.cancer.gov/ccg/research/genome-sequencing
COSMIC	Güncel mutasyon vt.	https://cancer.sanger.ac.uk/cosmic
cBioPortal	Kapsamlı veri içeren vt. portalı	https://www.cbioportal.org/datasets
GEO	Gen ekspresyon vt.	https://www.ncbi.nlm.nih.gov/gds
SomamiR	miRNA ve mutasyon vt.	https://compbio.uthsc.edu/SomamiR/
EGA	Genel veri deposu, büyük veri	https://ega-archive.org/datasets/
canSAR	Kapsamlı vt., ilaç keşfi	https://cansar.ai/data-sources
MethyCancer	DNA metilasyonu, gen/mutasyon vt.	http://methycancer.psych.ac.cn/
UCSC	Kapsamlı vt., büyük veri	https://genome.ucsc.edu/
MSKCC	Kanser mutasyon vt.	https://library.mskcc.org/find/database/a-z/all

CCLE	Kapsamlı vt., büyük veri	https://sites.broadinstitute.org/ccle/
GWAS	Genom düzeyinde ilişki verileri	https://gwas.mrcieu.ac.uk/datasets/
ARCHS4	Kapsamlı mRNA ekspresyon vt.	https://maayanlab.cloud/archs4/

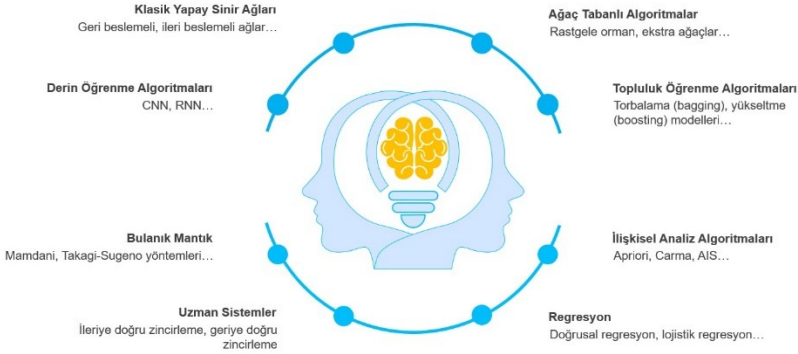
YAPAY ZEKA NEDİR?

Yapay zeka, insan zekasına özgü olan öğrenme, tespit etme, problem çözme, karar verme gibi faaliyetleri gerçekleştirebilen donanım ve yazılımlardır. Yapay zekâ, "koşula bağlı kurallar" olarak ortaya çıkmış, yıllar içinde insan beynine benzer şekilde çalışan daha karmaşık algoritmaları içerecek şekilde ilerlemiştir. Yapay zekâ teknolojisi, verileri analiz etmek için bilgisayarlı görü, derin öğrenme, makine öğrenimi, doğal dil işleme, robotik, konuşma, denetimli öğrenme ve denetimsiz öğrenme gibi uygulamaları kullanmaktadır. Bu uygulamalardan makine öğrenimi yapılandırılmış ve/veya yapılandırılmamış verilerden öğrenen, gizli kalıpları belirleyebilen, sınıflandırma yapabilen ve daha ileri adımları öngörebilen algoritmalarından oluşmaktadır ve yapay zekanın bir alt alanıdır (Tarı & Arpacı, 2024).

Derin öğrenme ise ses, görüntü ve video gibi kalıpları tanımlamak ve bu verileri işlemek için sinir ağlarından oluşan makine öğrenimi tabanlı bir yaklaşımdır. Makine öğrenimi, ham verilerden elde edilen özelliklerin manuel olarak oluşturulmasını ve sıralanmasını gerektirirken derin öğrenme, yüksek boyutlu ve heterojen olan ham verilerden en önemli özellikleri otomatik olarak öğrenebilecek katmanlara sahiptir. Ayrıca işlenecek verinin hacmi arttıkça ve veriler arasındaki ilişkiler karmaşıklaştıkça makine

öğrenimi yerini derin öğrenmeye bırakmaktadır. Çünkü derin öğrenme, yeterli miktarda veri olduğu müddetçe daha karmaşık görevleri daha verimli şekilde işleyebilmektedir.

Yapay zeka uygulamalarında yaygın olarak kullanılan algoritmalar Şekil 3'te görülmektedir.



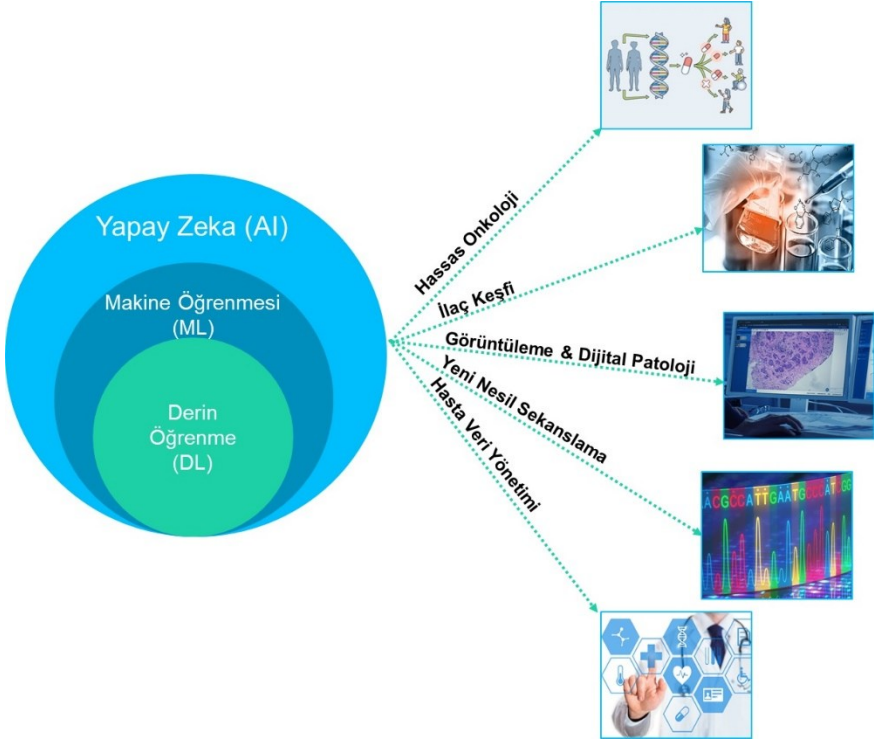
Şekil 3: Yapay zekâ uygulamalarında yaygın olarak kullanılan algoritmalar

KANSER GENOMİĞİNDE YAPAY ZEKA

Kapsamlı ve karmaşık verilerin artan kullanılabilirliği, kanser genomini, yapay zeka ile kanser araştırmaları yapanlar için çekici bir alan haline getirmiştir. İlgili odağı, insan genomunun sekans verilerinin yanı sıra, transkribe edilen genlerin veya çeşitli vücut hücreleri ve organlarında üretilen proteinlerin verileridir. Bunlar, diğerlerinin yanı sıra biyobankalardan veya elektronik hasta kayıtlarından alınan klinik verilerle birleştirilebilir. Yapay zekanın ve makine öğreniminin insan genomu ve biyotıpta kullanımı, genomdaki bireysel farklılıkların veya mutasyonların belirli özelliklere veya hastalıklara nasıl neden olduğuna dair yanıtlar elde

etme umuduyla ilişkilidir. Bu, genetik farklılıkların işlevsel sonuçlarına, teşhislere, tedavi seçeneklerine ilişkin prognozlara veya yeni ilaçların hızlı bir şekilde geliştirilmesine ilişkin tahminlere olanak sağlayacaktır (Heil & ark., 2021).

Bu kapsamda, yapay zekanın onkolojide kullanılan güncel uygulamalarından bazıları Şekil 4’te görülmektedir.



Şekil 4: Onkolojide yapay zeka uygulamaları (Dlamini & ark., 2020)

Kanserin Erken Tanısı, Teşhisi ve Kansere Evreleme

Erken teşhis birçok kanser türünde tedavi başarısını önemli ölçüde arttırmaktadır. Uluslararası alanda erken tanı, Dünya Sağlık

Örgütü (WHO) ve Uluslararası Kanser Erken Teşhis Birliği (ACED) dahil olmak üzere birçok kuruluş tarafından temel bir öncelik olarak kabul edilmektedir. Birçok çalışma, taramanın kanserin erken tespitini arttırarak mortalite oranlarını düşürebileceğini göstermektedir. Yapay zeka, taranan bireylerde klinik parametrelere göre araştırmayı veya sağlık kuruluşuna sevki tetikleyerek ve kapasitenin sınırlı olduğu durumlarda klinik iş akışlarını otomatikleştirerek kanser teşhisini doğrudan kolaylaştırma potansiyeline sahiptir (Hunter, Hindocha & Lee, 2022). Yapay zekanın bu amaçla kullanıldığı uygulama alanları Şekil 5'te verilmiştir.



Şekil 5: Kanser Erken Teşhisinde Yapay Zeka Uygulama Alanları

Kanser tanısı gen ifadesi, radyolojik görüntüler, histopatoloji veya bu verilerin bir kombinasyonu gibi ayrıntılı verilerin analizini gerektirir. 2000'li yılların başından beri yapay zeka, gen ekspresyon profilleri aracılığıyla kanser biyobelirteçlerini tespit etmek için kullanılmaktadır (Bhinder & ark., 2021).

Kanseri doğru bir şekilde teşhis eden ve kanser alt tiplerini doğrudan histopatolojik ve diğer tıbbi görüntülerden tanımlayan derin öğrenmeye dayalı modeller kapsamlı bir şekilde rapor edilmiştir. Derin Sinir Ağları (DNN) tümörü sağlıklı hücrelerden ayırmak için yüksek tahmin doğruluklarına ulaşırken, yakından ilişkili kanser alt tipleri arasında ayırım yapmak gibi daha zorlu sınıflandırma modelleri için de kullanılır. Örneğin Esteva ve arkadaşlarının etiketlenmiş cilt lezyonu görüntüleri üzerinde eğittikleri bir derin sinir ağı modeli, görüntülerin karsinom ve melanom sınıflandırması açısından test edildiğinde, dermatologlar tarafından yapılan sınıflandırma doğruluğu ortalamasından (AUC = 0,91-0,94) daha iyi performans göstermiştir (Esteva & ark., 2017). Benzer şekilde Jian ve arkadaşları, klinik ve patolojik özelliklerden elde edilene kıyasla daha iyi bir AUC (0.92-0.94) ile mide kanserlerinde gizli metastazı öngören bir derin sinir ağı modeli geliştirmek için bilgisayarlı tomografi görüntülerini kullanmışlardır (Jiang & ark., 2021). Fiziksel olarak herhangi bir sağlık kurumuna başvurmaksızın bireylerin kullanabileceği ve medikal görüntülerden kanserli lezyonların tespitine olanak veren mobil uygulamalar da geliştirilmiştir (Bhinder & ark., 2021).

Yapay zekanın kanser evreleme alanındaki uygulamaları da benzer şekilde 2000'li yılların başlarından bu yana mevcuttur. Kanserin evrenmesi ve derecelendirilmesi, yani kanserin ne kadar agresif ve ilerlemiş olduğunun belirlenmesi, teşhis sürecinin önemli bir bileşenidir. Yapay zeka ile kanser evrelemesi radyolojik görüntüler kullanılarak veya genomik profiller gibi görüntü dışı verilerle yapılabilmektedir. Örneğin biyopsi verileri eğitilerek

Gleason derecelendirme sistemine uygun şekilde prostat kanserinin hangi evrede olduđu tahmin edilebilmektedir (Kourou & ark., 2021).

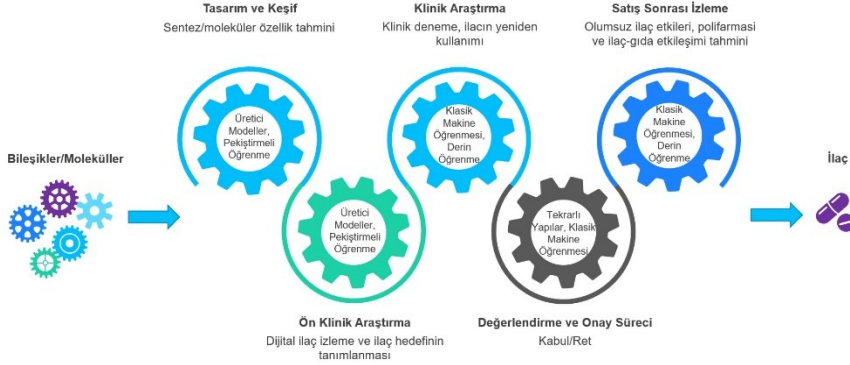
Tümör varlığının erken dönemde tespiti, kanser türünün teşhisi ve evrelemenin doğruluđu, tümör agresifliğinin temel belirleyicilerindedir ve klinik kararları ve prognozu etkiler. Yapay zeka, yalnızca birkaç on yıl içinde onkolojinin bu kritik alanına, bazen insan uzmanların performansı ile kıyaslanabilir şekilde ve ek olarak ölçeklenebilirlik ve otomasyon avantajıyla önemli katkılarda bulunmuştur (Bertsimas & Wiberg, 2020).

İlaç Keşfi

İlaç keşfi genomik, proteomik, klinik veriler ve mikrodizi verileri gibi çok büyük veri kümelerinin dahil olduđu, zaman alan ve karmaşık bir süreçtir. Bu alanlardaki fikir, enfeksiyon ajanlarının aktivitesini ve emilim, dağılım, metabolizma ve toksisite parametrelerini aynı anda dikkate alabilen çok ölçekli modeller geliştirmektir (Álvarez-Machancoses & ark., 2020).

Yapay zeka, ilaç keşfi ve aşı geliştirme alanlarında da güçlü bir araç olarak kullanılmaktadır. Yapay zeka dijital tarama, ilaç tasarımı ve optimizasyonu, peptid sentezi, ilacın yeniden konumlandırılması, görüntü analizi yoluyla hedef tanımlama ve doğrulama, farmakokinetik ile toksisite tahmini gibi çok sayıda ilaç keşif sürecinde uygulanmıştır (Şekil 6). Yapay zekanın ilaç keşfindeki bu uygulamaları, geleneksel ilaç geliştirme süreçleriyle ilişkili zamanı, maliyeti ve hata oranlarını önemli ölçüde azaltma potansiyeline sahiptir. Li ve ark. psödoefedrin andrografolid, klorokin, abakavir ve talidomid dahil olmak üzere COVID-19'a yönelik potansiyel tedavileri tanımlamak için yeni bir ağ tabanlı ilaç

yeniden kullanım platformu önermişlerdir (Li & ark., 2021). Bazı çalışmalarda ise, COVID-19 için potansiyel ilaçları tahmin etmek amacıyla protein-protein etkileşim ağları derin öğrenme modelleriyle birleştirilmiştir (Chafai & ark., 2023).



Şekil 6: İlaç tasarımı aşamaları ve bu aşamalarda sıklıkla kullanılan yapay zeka alt dalları

Kanserle İlişkili Sürücü Gen Tahmini

Kanser genom analizinin temel amacı, sürücü mutasyonları taşıyan kanser genlerinin tanımlanmasıdır. Ancak tümör hücrelerinde meydana gelen somatik mutasyonların çoğunun işlevsel bir etkisi yoktur. Mutasyonlar sürücüler ve taşıyıcılar olarak iki kategoriye ayrılabilir. Sürücü mutasyon, kanserin ortaya çıktığı dokuda pozitif seçilime olanak vererek kanser hücresine bir avantaj sağladığı için tümör oluşumunda nedensel olarak rol oynar. Herhangi bir işlevsel etkisi olmayan somatik mutasyonlar ise, hücre bölünmeleri sırasında sıklıkla meydana gelir ve miktar olarak sürücü mutasyonlardan epey fazladır. Sürücü mutasyonlara sahip genler sürücü genler olarak tanımlanırlar ve bu genler aynı zamanda taşıyıcı

mutasyonlara da maruz kalmaktadır. Dolayısıyla tümör oluşumuna neden olan sürücü genleri tespit etmek zordur (Teboul, 2022).

Bir kanser genomundaki sürücü genleri tanımlamak için çeşitli yapay zeka tabanlı yöntemler geliştirilmiştir. İlk yöntemler tümörün mutasyon yüküne, mutasyonların niteliksel özelliklerine (örneğin fonksiyonlarına) ve yoğunlaştıkları sıcak noktalara odaklanmıştır. Daha sonraları ağ tabanlı ve yola dayalı yöntemler de geliştirilmiştir. Bu konuda araştırmacılar tarafından, yapay zekanın hem sınıf etiketlerinin mevcut olduğu denetimli öğrenme algoritmaları (genlerin hangi kategoriye dahil olduğunun önceden bilindiği durumlar) hem de denetimsiz öğrenme algoritmaları (genlerin hangi kategoriye dahil olduğunun bilinmediği durumlar) kullanılmıştır. Araştırmacıların bazıları yöntemini tek bir tür veri üzerinde eğitmiş bazıları ise farklı veri türlerini entegre şekilde kullanmış ve başarılı sonuçlar elde etmişlerdir. Bunların önemli bir kısmı genomik ve epigenomik veri entegrasyonuna dayanmaktadır. Örneğin mutasyon verileri ile DNA metilasyon verilerinin ya da mutasyon verileri ile ekspresyon verilerinin aynı anda kullanımına sıkça rastlanmaktadır.

Akademik ve klinik çalışmaların sonuçlarına dayanarak oluşturulan ve düzenli olarak güncellenen genel kabul görmüş bazı sürücü gen veritabanları da mevcuttur. COSMIC (Cancer Gene Census) (Sondka & ark., 2018) ve NCG (The Network of Cancer Genes) (Dressler & ark., 2022) bunların en önemlilerindedir. Ancak sürücü olmayan genlerin kesin bir listesi henüz bulunmadığından geliştirilen yöntemlerin sonuçları, yanlış negatifler bağlamında şeffaf olarak değerlendirilememektedir.

Bir sürücü genin karsinogeneze katkıda bulunması iki şekilde olabilmektedir: 1) Tümörün tutunmasını ve yayılmasını kolaylaştıracak aktivitelerle ve 2) hücre içinde yolunda gitmeyen olay ve durumlar karşısında sessiz kalarak. Birinci durumdaki sürücüler onkogen, sessizleştirilmiş sürücüler ise tümör baskılayıcı gen ismini almaktadır. Bu ayrımın yapılması, o sürücü genin/proteinin hedef alındığı tedavi planlarının daha başarılı olmasını sağlayabilir. Ancak bazı sürücü genlerin her iki işleve de sahip olabildiği çeşitli çalışmalarda gösterilmiştir.

Kanser Gen/Protein Etkileşim Ağları

Kanser, birden fazla gen veya faktörün etkilendiği biyolojik süreçlerde meydana gelen karmaşık bir hastalıktır. Gen-gen etkileşimlerinin modellenmesi, altta yatan biyolojik mekanizmaların anlaşılmasına yardımcı olur. Çünkü peptitler ve gen ürünleri olan proteinler sosyal moleküllerdir, birçok biyolojik süreçte birlikte hareket etmektedirler (Ye & ark., 2023).

Geleneksel istatistiksel araçlar, büyük ölçekli genetik verilerin analizi için çok uygun değildir. Bununla birlikte, gen-gen etkileşimlerini tespit etmeye yönelik bazı hesaplama sınırlamalarının, makine öğrenimi ve veri madenciliği gibi modern teknikler kullanılarak aşılabileceği görülmektedir. Çoklu genler arasındaki etkileşimleri tespit etmek, kombinatorial bir optimizasyon problemi olarak düşünülürse, problem, belirli bir veri kümesinden en yüksek tahmin doğruluğunu üretebilecek en iyi gen kombinasyonlarının bulunması olarak tanımlanabilir (Huang & ark., 2018). Bu kapsamda sekans tabanlı, yapı tabanlı, genomik özellik tabanlı, ağ topoloji tabanlı ve alan/motif tabanlı tahmin modelleri

geliştirilmiştir. Ek olarak filogenetik ağaç, gen füzyonu ve ekspresyon bilgilerini kullanan yaklaşımlar da mevcuttur.

Bu amaca yönelik olarak yapay zeka modellerinde kullanılabilir olan protein/peptit etkileşim veritabanı kaynaklarının bir listesi Tablo 2’de verilmiştir.

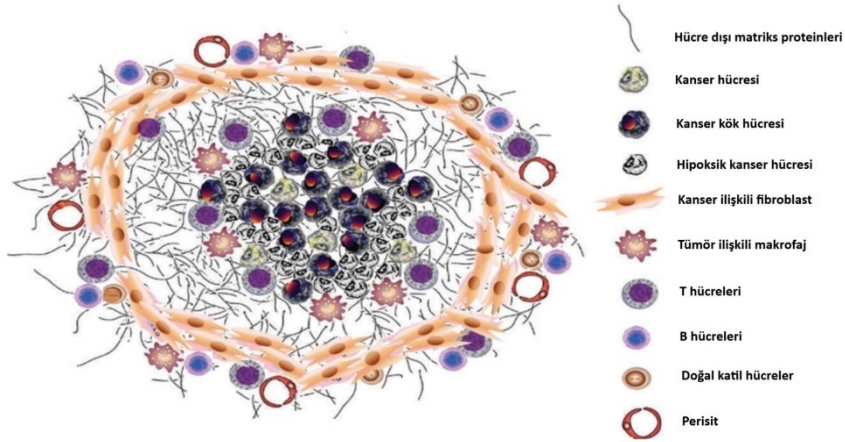
Tablo 2: Yapay zeka modellerinde kullanılabilir protein/peptit etkileşim veritabanı kaynakları (Ye & ark., 2023)

Veritabanı Adı	Temel Özellikler	Web Erişim Linki
BioGRID	Proteinlerin genetik ve kimyasal etkileşimlerini içeren vt.	https://thebiogrid.org/
UniProt	Gen sekanslarını ve açıklamalarını içeren bir vt.	https://www.uniprot.org/
STRING	Bilinen/tahmin edilen etkileşimleri içeren kapsamlı bir vt.	https://string-db.org/
IntAct	Literatüre ve direkt kullanıcı verilerine dayanan etkileşim vt.	https://www.ebi.ac.uk/intact/home
MINT	Deneyssel olarak kanıtlanmış protein etkileşimlerini içeren vt.	https://mint.bio.uniroma2.it/

Tümör Mikroçevresinin Karakterizasyonu

Kanser hücreleri, tümör gelişimi ve metastaz için heterojen yapıda olan tümör mikroçevresine ihtiyaç duymaktadırlar. Tümör mikroçevresinde yer alan bileşenler ile karşılıklı iletişim halinde olan kanser hücreleri, bu iletişim sayesinde metastazını kolaylaştıracak proliferatif sinyallerin sürdürülmesini sağlayarak

hücre ölümüne direnç, anjiogenezin indüksiyonu ve bağışıklık hücrelerinin yıkıcı etkisinden kaçınma gibi özellikler edinmektedirler (Baş Topcu, 2022) (Şekil 7).



Şekil 7: Tümör mikroçevresinde yer alan bileşenler (Eskiizmir & Özgür, 2018)

Tümör mikroçevresinde bulunan bileşenler, tümör hücrelerinin bağışıklık hücreleri tarafından yok edilmesini önlemek için bağışıklık baskılayabilmektedir. Ayrıca, tümörün büyümesi ve ilerlemesine uyum sağlamak ve kemoterapi gibi herhangi bir stres durumuna karşı koymak için dinamik olarak uyarlanabilmektedir. Bu nedenlerden dolayı tümör mikroçevresini hedeflemenin kanser tedavisi için etkili bir yol olacağı düşünülmektedir (Baş Topcu, 2022). Bu kapsamda geliştirilen tedavi yöntemleri tümör mikroçevre bileşenlerinin sentezinin önlenmesini, stroma hücrelerinin sentezlediği proteinlerin hedeflenmesini, kanser hücresi ile stroma arası etkileşimlerin engellenmesini ve Kimerik Antigen Reseptör T hücreleri denilen genetik olarak modifiye edilmiş bağışıklık sistemi

hücreleri ile stromal antijenlerin hedeflenmesini içermektedir. Bu tedaviler sitotoksik kanser ilaçları denilen klasik kemoterapi ilaçları ile birlikte denenmektedir. Böylece ilaç direncinin ve dolayısıyla kullanılacak ilaç dozlarının azalması sağlanabilecektir.

Tümör analizini genişleten yapay zeka yaklaşımları, tümör mikroçevresinin konumsal ve niceliksel değerlendirmesi için kullanılmaktadır. Bu tür bir karakterizasyon, kontrol noktası immünoterapileri bağlamında tümör ile bağışıklık sistemi arasındaki mekanizma bozukluğunun anlaşılması için özellikle önemlidir (Bhinder & ark., 2021). Tümör mikroçevresinin değerlendirilmesine yönelik mevcut yöntemler, geleneksel histopatolojik gözlem, periferik kan testi, ilgili gen analizi ve radyoloji görüntülerinin analizlerini içerebilir (Lin & ark., 2023).

Histopatolojik görüntü verileri kullanılarak eğitilen ve uzman patolog değerlendirme sonuçlarıyla büyük oranda örtüşen yapay zeka modelleri literatürde yer almaktadır (Nagpal & ark., 2019; Zhou & ark., 2019). Özellikle derin öğrenme algoritmalarının kullanımı artmaktadır. Ancak doğal dil işleme ve Rastgele Orman gibi klasik makine öğrenimi algoritmaları da çeşitli çalışmalarda uygulanmıştır. Bu çalışmalarda, görüntü verilerinin yanı sıra genomik veriler de (RNA sekansları veya mikrodizi verileri gibi) kullanılmıştır. Çünkü Yeni Nesil Sekanslama ve tek hücreli dizilemenin artmasıyla birlikte tümör mikroçevresinin, ekspresyonu değişmiş genlere dayalı bir analiz modeli ile de değerlendirilebileceği anlaşılmıştır. Ancak, daha net biyolojik anlamların araştırılmasına ve dolayısıyla daha fazla çalışmaya ihtiyaç vardır. Ayrıca bulguların deneysel olarak test edilmesi gerekmektedir (Lin & ark., 2023).

SONUÇ

Yapay zekanın; kanserin önlenmesi, teşhisi, tedavisi ve kanser hastası bakımında tartışılmaz bir potansiyeli vardır. Özellikle sınıflandırma ve tahmin uygulamalarında insan uzmanlardan daha iyi öngörüler sunması giderek yaygınlaşmaktadır. Ancak henüz sağlık kurumlarında ve sağlık çalışanları tarafından kullanımı tam olarak benimsenmemiştir. Üretilen modellerin yorumlanabilirliğini artırmaya yönelik ek araştırmalar, yapay zekanın klinik kullanımını daha kabul edilebilir hale getirecektir. Ayrıca onkoloji ile mühendislik alanında daha fazla işbirliğinin sağlanması kanser genomiğinde inovatif çalışmaları arttırabilecektir.

Çeşitli kanser veri kümelerinin yeterince homojen olmaması, veritabanlarında genellikle beyaz tenli Avrupalı ve Amerikalı hastaların örneklerinin sunulması bir diğer problemdir. Araştırmalar farklı coğrafyalarda yaşayan farklı etnik toplulukların, hastalıkların insidansı ve mortalitesi bakımından farklı özellikler gösterdiğini ortaya koymuştur. Dolayısıyla homojen olmayan verilerle eğitilen modellerin farklı hasta profillerinde iyi sonuçlar üretmemeye potansiyeli mevcuttur.

Kanser genetik düzeydeki değişikliklerle ilgili olsa da her hastalıkta olduğu gibi bazı risk faktörleri tetikleyici olmaktadır. Kanser risk faktörleri sadece genetik yatkınlık değildir, aynı zamanda çevresel risk faktörleri, yaşam biçimi ve beslenme tarzı da tümör oluşumunda etkili olabilmektedir. Kişiselleştirilmiş kanser tedavisinin yaygınlık kazandığı günümüzde hastalara ait farklı verilerin düzenli olarak toplanması ve analizi kanserin önlenmesinde erken uyarı sistemi sunma potansiyeline sahiptir. Hatta yapay zeka tabanlı mobil uygulamalar aracılığıyla toplumdaki bireylerin kanser

riskleri raporlanabilir ve böylece farkındalığın arttırılması sağlanabilir.

KAYNAKÇA

Álvarez-Machancoses, Ó., DeAndres Galiana, E. J., Cernea, A., Fernández de la Viña, J., & Fernández-Martínez, J. L. (2020). On the role of artificial intelligence in genomics to enhance precision medicine. *Pharmacogenomics and personalized medicine*, 105-119. <https://doi.org/10.2147/PGPM.S205082>

Arjmand, B., Hamidpour, S. K., Tayanloo-Beik, A., Goodarzi, P., Aghayan, H. R., Adibi, H., & Larijani, B. (2022). Machine learning: a new prospect in multi-omics data analysis of cancer. *Frontiers in Genetics*, 13, 824451. <https://doi.org/10.3389/fgene.2022.824451>.

Baş Topcu, K.S. (2022). Tümör ilerlemesinde tümör mikroçevrenin rolü. *Türk Nöroşirurji Dergisi*, 32(1), 98-104.

Baykara, O. (2016). Kanser Tedavisinde Güncel Yaklaşımlar. *Balıkesir Sağlık Bilimleri Dergisi*, 5(3), 154-165. <https://doi.org/10.5505/bsbd.2016.93823>.

Belciug, S. (2020). *Artificial intelligence in cancer: diagnostic to tailored treatment*. Academic Press.

Bertsimas, D., & Wiberg, H. (2020). Machine learning in oncology: methods, applications, and challenges. *JCO clinical cancer informatics*, 4. <https://doi.org/10.1200/CCI.20.00072>.

Bhinder, B., Gilvary, C., Madhukar, N. S., & Elemento, O. (2021). Artificial intelligence in cancer research and precision medicine. *Cancer discovery*, 11(4), 900-915. <https://doi.org/10.1158/2159-8290.CD-21-0090>.

Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R., Ozenberger, B. A., Ellrott, K., ... & Stuart, J. M. (2013). The cancer genome atlas pan-cancer analysis project. *Nature genetics*, *45*(10), 1113-1120.

Caudai, C., Galizia, A., Geraci, F., Le Pera, L., Morea, V., Salerno, E., ... & Colombo, T. (2021). AI applications in functional genomics. *Computational and Structural Biotechnology Journal*, *19*, 5762-5790. <https://doi.org/10.1016/j.csbj.2021.10.009>.

Chafai, N., Bonizzi, L., Botti, S., & Badaoui, B. (2024). Emerging applications of machine learning in genomic medicine and healthcare. *Critical Reviews in Clinical Laboratory Sciences*, *61*(2), 140-163. <https://doi.org/10.1080/10408363.2023.2259466>.

Eskiizmir, G., & Özgür, E. (2018). Epithelial-mesenchymal transition in tumor microenvironment induced by hypoxia. *Cancer Metastasis*, 7-24.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature*, *542*(7639), 115-118.

Grzybowski, A., Pawlikowska-Łagód, K., & Lambert, W. C. (2024). A history of artificial intelligence. *Clinics in Dermatology*, *42*(3), 221-229. <https://doi.org/10.1016/j.clindermatol.2023.12.016>

Heil, R., Heyen, N. B., Baumann, M., Hüsing, B., Bachlechner, D., Schmoch, U., & König, H. (2021). Artificial

intelligence in human genomics and biomedicine.
<https://doi.org/10.14512/tatup.30.3.30>.

Huang, S., Cai, N., Pacheco, P. P., Narrandes, S., Wang, Y., & Xu, W. (2018). Applications of support vector machine (SVM) learning in cancer genomics. *Cancer genomics & proteomics*, *15*(1), 41-51. <https://doi.org/10.21873/cgp.20063>.

Hunter, B., Hindocha, S., & Lee, R. W. (2022). The role of artificial intelligence in early cancer diagnosis. *Cancers*, *14*(6), 1524. <https://doi.org/10.3390/>.

Kourou, K., Exarchos, K. P., Papaloukas, C., Sakaloglou, P., Exarchos, T., & Fotiadis, D. I. (2021). Applied machine learning in cancer research: A systematic review for patient diagnosis, classification and prognosis. *Computational and Structural Biotechnology Journal*, *19*, 5546-5555. <https://doi.org/10.1016/j.csbj.2021.10.006>.

Li, F., Michelson, A. P., Foraker, R., Zhan, M., & Payne, P. R. (2021). Computational analysis to repurpose drugs for COVID-19 based on transcriptional response of host cells to SARS-CoV-2. *BMC medical informatics and decision making*, *21*, 1-13.

Liao, J., Li, X., Gan, Y., Han, S., Rong, P., Wang, W., ... & Zhou, L. (2023). Artificial intelligence assists precision medicine in cancer treatment. *Frontiers in oncology*, *12*, 998222. <https://doi.org/10.3389/fonc.2022.998222>.

Lin, G., Wang, X., Ye, H., & Cao, W. (2023). Radiomic Models Predict Tumor Microenvironment Using Artificial Intelligence—the Novel Biomarkers in Breast Cancer Immune

Microenvironment. *Technology in Cancer Research & Treatment*, 22, 15330338231218227. <https://doi.org/10.1177/15330338231218227>.

Nagpal, K., Foote, D., Liu, Y., Chen, P. H. C., Wulczyn, E., Tan, F., ... & Stumpe, M. C. (2019). Development and validation of a deep learning algorithm for improving Gleason scoring of prostate cancer. *NPJ digital medicine*, 2(1), 48.

Pathak, Y., Saikia, S., Pathak, S., Patel, J., & Prajapati, B. G. (Eds.). (2023). *Artificial Intelligence in Bioinformatics and Chemoinformatics*. CRC Press.

Perez-Lopez, R., Ghaffari Laleh, N., Mahmood, F., & Kather, J. N. (2024). A guide to artificial intelligence for cancer researchers. *Nature Reviews Cancer*, 1-15. <https://doi.org/10.1038/s41568-024-00694-7>.

Sondka, Z., Bamford, S., Cole, C. G., Ward, S. A., Dunham, I., & Forbes, S. A. (2018). The COSMIC Cancer Gene Census: describing genetic dysfunction across all human cancers. *Nature Reviews Cancer*, 18(11), 696-705. <https://doi.org/10.1038/s41568-018-0060-1>.

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 71(3), 209-249.

Tarı, Ö., & Arpacı, N. (2024). İLAÇ TASARIMINDA YAPAY ZEKÂ UYGULAMALARI. *Journal of Faculty of*

Pharmacy of Ankara University, 48(1), 327-365.
<https://doi.org/10.33483/jfpau.1327078>.

Teboul, R. (2022). *Predicting cancer driver mutation with deep learning* (Doctoral dissertation, Université Paris Cité).

Zhou, Q., Zhou, Z., Chen, C., Fan, G., Chen, G., Heng, H., ... & Dai, Y. (2019). Grading of hepatocellular carcinoma using 3D SE-DenseNet in dynamic enhanced MR images. *Computers in biology and medicine*, 107, 47-57.

BÖLÜM III

Türkçe Metinlerde Anahtar Kelime Çıkarımı: LDA, RAKE ve BERTurk Yöntemlerinin Karşılaştırmalı Analizi

Kadir TOHMA¹
Halil İbrahim OKUR²

Giriş

Anahtar kelime çıkarımı, doğal dil işleme (DDİ) ve bilgi erişimi alanlarında kritik bir öneme sahiptir. Bu görev, arama motorlarından e-ticaret platformlarına, haber ve sosyal medya analizlerine kadar geniş bir uygulama yelpazesinde kritik bir rol oynar. Anahtar kelime çıkarımı, metinlerin temel kavramlarını ve odak noktalarını belirleyerek, bu metinlerin analiz edilmesini ve işlenmesini kolaylaştırır. Özellikle Türkçe metinlerde, dilin yapısal

¹ Dr. Öğr. Üyesi, İskenderun Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü, İskenderun/Türkiye, Orcid: 0000-0002-2631-7810, kadir.tohma@iste.edu.tr

² Dr., İskenderun Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü, İskenderun/Türkiye, Orcid: 0000-0003-0339-4626, hibrahim.okur@iste.edu.tr

karmaşıklığı ve morfolojik zenginliği nedeniyle, doğru ve etkili anahtar kelime çıkarımı önemli bir zorluk teşkil etmektedir (Tohma ve ark., 2020). Bu bağlamda, LDA (Latent Dirichlet Allocation) (Blei, D ve ark., 2003), RAKE (Rapid Automatic Keyword Extraction) (Rose, S. ve ark., 2010) ve BERTurk BERTurk (Schweter, S. 2020) gibi yöntemlerin karşılaştırmalı analizi, bu yöntemlerin performansını ve uygunluğunu değerlendirmek için gereklidir.

Bu çalışmada, Türkçe metinlerde anahtar kelime çıkarımı için yaygın olarak kullanılan bu üç yöntemin etkinlikleri, kapsamlı bir şekilde incelenmiştir. Kùltürler arası iletişim, dilin evrimi ve Doğal Dil İşleme alanının gelişimi gibi derinlemesine konuları ele alan zengin içerikli bir Türkçe metin üzerinde LDA, RAKE ve BERTurk yöntemleri uygulanmış ve karşılaştırılmıştır. Çalışmanın amacı, bu yöntemlerin Türkçe metinlerde ne ölçüde etkili olduğunu ve hangi senaryolarda en iyi performansı gösterdiğini belirlemektir.

Literatürde, farklı yöntemlerin çeşitli metin türlerinde ve uzunluklarında nasıl performans gösterdiğine dair önemli bulgular yer almaktadır. Nadim ve arkadaşlarının (2023) çalışmasında, KPminer'in farklı metin uzunluklarında en tutarlı performansı gösterdiği, KeyBERT yönteminin diğer araçlardan daha iyi performans gösterdiği ve RAKE ile YAKE'in (Campos, R., ve ark. 2020) yürütme süresi açısından en hızlı araçlar olduğu belirtilmiştir. Ayrıca, Ayan ve arkadaşlarının (2021) araştırmasında, Türkçe web sayfalarından anahtar kelime çıkarımı için BERT tabanlı bir yöntem olan Sentence-BERT'in etkileyici sonuçlar verdiği ve şirketlerin faaliyet alanları hakkında değerli ipuçları sunduğu gözlemlenmiştir. Erdağı ve arkadaşlarının (2023) çalışmasında ise, Türkçe haber

metinlerinde majör ünlü uyumu ve minör ünlü uyumu özelliklerinin, BERTurk modeli ile yakın sonuçlar vererek çıkarımsal özetleme sonuçlarını iyileştirdiği rapor edilmiştir. Ayrıca, Campos ve arkadaşlarının (2018) çalışmasında, RAKE'in tek belgelerden anahtar sözcük çıkarımında, TextRank (Mihalcea, R., ve ark., 2004) ve SingleRank gibi yöntemlerle karşılaştırıldığında daha iyi performans gösterdiği belirtilmiştir. Krivorogov ve arkadaşlarının (2023) çalışmasında, RAKE yönteminin öğrenci geri bildirimlerinde %79 ile en yüksek F1 BERTScore'a sahip olduğu gösterilmiştir. Issa ve arkadaşlarının (2023) çalışmasında ise, Paraphrase-mpnet-base-v2 modelinin uzun metinlerde anahtar kelime çıkarımında etkinlik ve verimlilik açısından en iyi sonuçları sağladığı gözlemlenmiştir. Bu literatür örnekleri, anahtar kelime çıkarımı yöntemlerinin farklı metin türleri ve uzunluklarında nasıl performans gösterdiğini geniş bir perspektiften ele almaktadır. Bu çalışmalar, çeşitli yaklaşımların güçlü ve zayıf yönlerini ortaya koyarak, araştırma ve uygulama alanlarında en etkili yöntemlerin belirlenmesine yardımcı olmaktadır. Bu bağlamda, Türkçe metinlerde anahtar kelime çıkarımı için en uygun yöntemin seçilmesi, uygulamanın amacına ve metin özelliklerine göre değişkenlik göstermektedir. Bu alandaki mevcut literatür, gelecekteki çalışmalar için önemli bir referans noktası oluşturarak, farklı yöntemlerin kapsamlı değerlendirilmesine ve daha etkili anahtar kelime çıkarım stratejilerinin geliştirilmesine katkıda bulunacaktır (Göz, F., ve ark., 2021, Eryiğit, G., ve ark., 2015, Ozdemir, B., ve ark., 2009).

Bu çalışmada ise, LDA, RAKE ve BERTurk yöntemlerinin performansları, özgün bir Türkçe metin üzerinde karşılaştırmalı olarak değerlendirilmiştir. Elde edilen sonuçlar, Türkçe metinlerde

anahtar kelime çıkarımı yapmak isteyen arařtırmacılar için önemli rehberlik sunmaktadır. Bu çalıřma, farklı uygulama alanlarında en uygun anahtar kelime çıkarım aracını seçmede yol gösterici olacaktır.

Materyal ve Yöntem

Bu çalıřmada, Türkçe metinlerde anahtar kelime çıkarımı için yaygın olarak kullanılan üç yöntem olan LDA, RAKE ve BERTurk yöntemlerinin performansları karşılaştırılmıřtır. Çalıřmanın amacı, bu yöntemlerin doğruluk, F1 skoru ve geri çağırma oranı gibi metrikler açısından etkinliğini deęerlendirmektir. Bu amaçla, özgün bir Türkçe metin (Şekil 1) üzerinde her bir yöntem uygulanmış ve sonuçlar karşılaştırılmıřtır. Çalıřmada kullanılan metin, dilin evrimi, kültürler arası iletişim ve Doğal Dil İşleme alanının gelişimi gibi konuları ele alan bir içerikten oluşmaktadır. Metin, kelimelere ayrıldıktan sonra Türkçe dilindeki gereksiz sözcükler (stopwords) filtrelenmiş ve metin temizlenmiştir. Bu süreçte, her kelimenin küçük harfe dönüřtürülmesi ve alfasayısal olmayan karakterlerin çıkarılması gibi işlemler gerçekleştirilmiştir. Bu adımlar, metni LDA, RAKE ve BERTurk yöntemleri ile analiz etmek için gerekli ön işleme adımlarını oluşturmuştur.

İnsanlık tarihinde dil önemli bir yer tutmaktadır. İnsanlar dil aracılığı ile birbirleriyle iletişim kurabilmiş ve anlaşabilmişlerdir. Diller zamanla gelişmiş ve hâlâ gelişmektedirler. Dillere yeni sözcükler eklenmekte ve bazı sözcükler unutulmaktadır. Ancak dillerin yapıları ve kuralları uzun süreler değişmezliğini korurlar. Birleşmiş Milletler verilerine göre günümüzde 4000'den fazla dilin konuşulduğu anlaşılmaktadır. Ancak bu dillerden bazıları bir avuç insan tarafından konuşulmaktadır. Dillerin gelişmişliği ile o dili konuşan toplumların kültür, sanat ve bilim alanında ürettikleri arasında sıkı bir bağlantı olduğu açıktır. Bilişim teknolojilerindeki gelişmeler, bilim insanlarını diller üzerinde çalışmaya teşvik etmiştir. İlk dönemlerde bilgisayar ile konuşmayı amaçlayan bilim insanları zamanla, konuşulanları veya yazılanları anlamaya ve değerlendirmeye çalışmıştır. Doğal Dil İşleme adı verilen bu bilim dalı, genel olarak, dillerin bilgisayar yardımıyla işlenmesi üzerinde çalışmaktadır. Bu çalışmaların sonunda, bir konuşma veya yazının anlaşılması, sorulara yanıt verilmesi ve diller arası çeviriler hedeflenmektedir. Bu hedeflere ulaşabilmek için dillerin ses, biçim, dizilim, anlam gibi temel konuları bu bağlamda incelenmektedir. Bu yazıda DDİ kapsamında yer alan temel konular, dillere ilişkin özellikler ve bazı DDİ uygulamaları tanıtılmıştır.

Şekil 1. Çalışmada kullanılan özgün metin

A.LDA Yöntemi

LDA (Latent Dirichlet Allocation) (Blei, D ve ark., 2003), belgeleri oluşturan gizli konuları modellemek amacıyla kullanılan olasılıksal bir modeldir. Bu yöntem, her bir belgenin birden fazla konudan oluştuğunu ve her konunun belirli kelimelerle ilişkili olduğunu varsayar. LDA, büyük metin koleksiyonlarında ortak temaları veya konuları keşfetmek ve belgelerin bu konularla nasıl ilişkilendirildiğini anlamak için yaygın olarak kullanılır. Model, belgelerdeki kelimelerin dağılımını analiz ederek, her bir belgenin hangi konulara ne ölçüde bağlı olduğunu tahmin eder. Bu bağlamda, LDA, belge kümeleri içindeki gizli yapıları ortaya çıkarmada etkili bir araçtır.

Ancak, Türkçe gibi eklemeli (aglutinatif) ve yapısal olarak zengin dillerde, LDA'nın etkin bir şekilde çalışabilmesi için bazı ek adımların uygulanması gerekir. Türkçe'nin eklemeli yapısı, kelime köklerine eklenen çeşitli gramatik ekler yoluyla yeni kelimeler

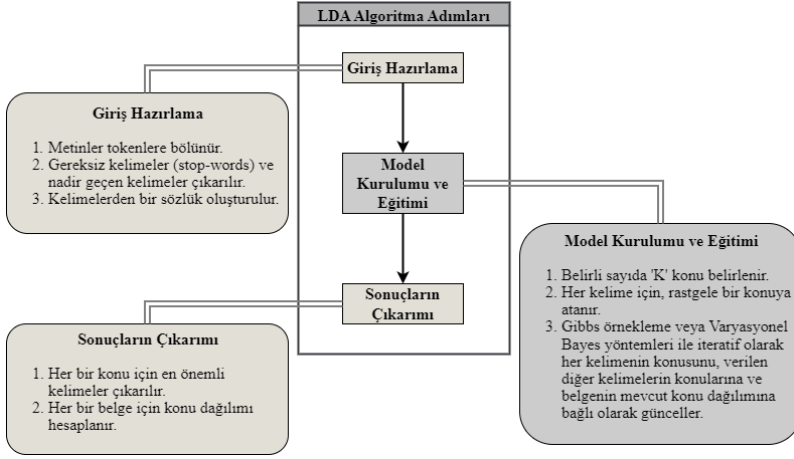
türetilmesine olanak tanır. Bu durum, kelimelerin biçimsel olarak farklılaşmasına ve anlamlarının değişmesine neden olabilir, bu da LDA modelinin kelimeleri ve konuları doğru bir şekilde eşleştirmesini zorlaştırabilir. Türkçe’de aynı kelimenin farklı eklerle türetilmiş birçok varyasyonu olabilir; bu da modelin kelimeleri tekil biçimlerine ayırmasını ve bu biçimler üzerinden analiz yapmasını gerektirir.

Türkçe metinlerde LDA’nın daha başarılı sonuçlar verebilmesi için, kelimelerin kök ve eklerinin doğru bir şekilde ayrıştırılması kritik bir önem taşır. Bu nedenle, LDA uygulamasından önce metinlerin morfolojik analiz araçlarıyla işlenmesi gereklidir. Morfolojik analiz, kelimelerin köklerine ve gramatik eklerine ayrılmasını sağlar ve böylece modelin dilin yapısını daha iyi anlamasına yardımcı olur. Türkçe’nin eklemeli yapısı, kelime köklerinin çeşitli eklerle birleşip farklı anlamlara bürünmesine neden olduğundan, LDA’nın doğru sonuçlar verebilmesi için bu ayrışmanın doğru bir şekilde yapılması, konuların daha net bir şekilde belirlenmesini sağlar.

Bu işlem sayesinde, LDA modeline girilen metinlerdeki kelimeler, kök hallerine indirilir ve bu sayede model, benzer anlamlara sahip kelimeleri bir araya getirerek daha tutarlı ve anlamlı konu dağılımları çıkarabilir. Örneğin, "çalışıyor", "çalıştı" ve "çalışacak" gibi kelimeler, morfolojik analiz sonrasında "çalış-" köküne indirgenecektir. Bu, LDA’nın bu kelimeleri aynı konuya dahil etmesini ve konunun dilde nasıl ifade edildiğine dair daha bütüncül bir anlayış geliştirmesini sağlayacaktır.

Türkçe metinler için LDA modelinin başarısını artırmak amacıyla, kelime öbeklerinin de dikkatli bir şekilde ele alınması gerekebilir. Bu bağlamda, sadece kelime köklerinin ayrıştırılması değil, aynı zamanda kelime öbeklerinin model tarafından nasıl algılandığı da önem taşır. Kelime öbeklerinin anlamını doğru bir şekilde modelleyebilmek için, bu öbeklerin hangi konularla ilişkilendirildiği ve bu öbeklerdeki kelimelerin nasıl bir araya geldiği analiz edilmelidir. Bu, modelin daha derin bir dilsel anlayış geliştirmesine yardımcı olur.

Sonuç olarak, LDA'nın Türkçe gibi eklemeli dillerde etkili bir şekilde uygulanabilmesi için, dilin morfolojik zenginliği dikkate alınmalı ve kelime kökleri ile ekler arasındaki ilişkiler doğru bir şekilde analiz edilmelidir. Türkçe metinlerde morfolojik analiz araçlarının kullanılması, modelin daha tutarlı ve anlamlı konu dağılımları tahmin etmesine olanak tanır. Bu sayede, LDA, Türkçe metinlerdeki konuların daha net ve doğru bir şekilde tespit edilmesini sağlayarak, dil işleme süreçlerine önemli katkılarda bulunabilir (Şekil 2).



Şekil 2. LDA Algoritma Adımları

B.RAKE Yöntemi

RAKE (Rapid Automatic Keyword Extraction) yöntemi, Rose ve arkadaşları (2010) tarafından geliştirilmiş olup, metin içindeki önemli kelime öbeklerini ve bu öbeklerin metindeki ağırlıklarını belirlemek amacıyla istatistiksel ve dil bilimsel yöntemler kullanır. Bu yöntem, özellikle uzun ve karmaşık metinlerde, anahtar kelimelerin ve kelime gruplarının hızlı bir şekilde çıkarılmasını sağlamak için tercih edilir. Ancak, her dilin kendine özgü dilbilgisel ve yapısal özellikleri bulunduğu için, RAKE yönteminin uygulanacağı dilin yapısına uygun özelleştirmeler yapılması gereklidir. Türkçe gibi eklemeli dillerde, kelime öbeklerinin doğru bir şekilde tespit edilmesi ve anlamlı sonuçlar elde edilmesi için yöntemin bazı dil spesifik düzenlemelerle uyarlanması önemlidir.

Türkçe'nin eklemeli yapısı, kelime köklerine ekler eklenerek yeni kelimeler ve anlamlar oluşturulmasını sağlar. Bu durum, kelime öbeklerinin çıkarımında belirli zorluklar ortaya çıkarabilir, çünkü eklemeli yapılar, kelimelerin morfolojik olarak değişmesine ve farklı

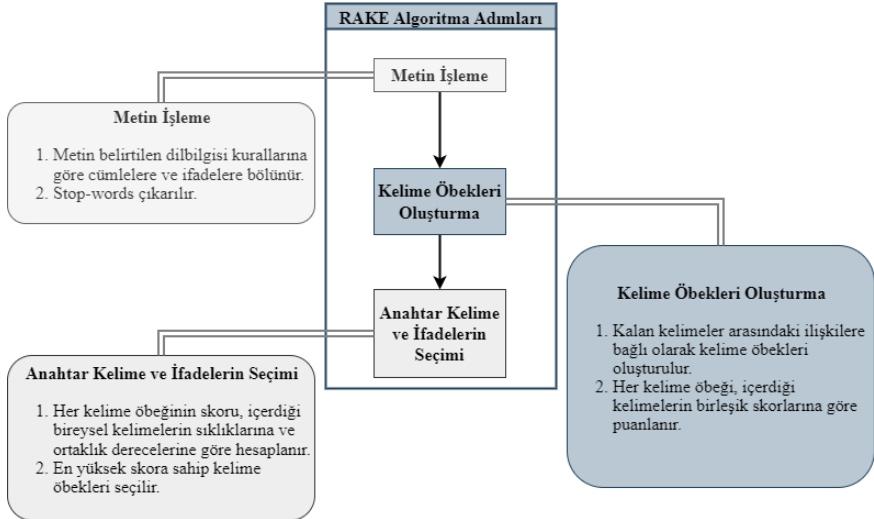
anamlara gelmesine neden olabilir. Bu nedenle, Türkçe metinlerde RAKE yönteminin uygulanabilmesi için, dilin yapısına uygun özel düzenlemeler yapılmalıdır. Örneğin, Türkçe'de sıkça kullanılan bağlaçlar, edatlar, zamirler ve diğer işlevsel kelimeler, metin analizinde çıkarılmalıdır. Bu tür işlevsel kelimeler, kelime öbeklerinin önemini belirlerken yanıltıcı olabilir ve anlamlı sonuçlar elde edilmesini zorlaştırabilir. Bu bağlamda, RAKE yönteminin Türkçe'ye uyarlanması, stop-words olarak bilinen bu işlevsel kelimelerin önceden belirlenip filtrelenmesini gerektirir.

Bunun yanı sıra, Türkçe'nin eklemeli yapısı göz önüne alındığında, kelime öbekleri içindeki kelimeler arasındaki morfolojik ve semantik bağlantıların analiz edilmesi de büyük bir önem taşır. Kelimelerin sadece birlikte kullanım sıklıkları değil, aynı zamanda bu kelimeler arasındaki anlam ilişkileri de dikkate alınmalıdır. Türkçe'deki eklemeli yapı, kelime öbekleri içinde bulunan kelimelerin birbirine nasıl bağlandığını ve hangi anlamları taşıdığını etkileyebilir. Bu nedenle, RAKE yönteminin Türkçe uygulamalarında, bu tür bağlantıları tespit edebilecek özgün doğal dil işleme (DDİ) tekniklerinin entegrasyonu zorunludur. Bu teknikler, kelimelerin hangi eklerle birleştiğini, hangi bağlamlarda kullanıldığını ve öbek içindeki kelimelerin birlikte kullanımının ne kadar anlamlı olduğunu değerlendirir.

Örneğin, RAKE yönteminin Türkçe'de uygulanması sırasında, "yapmak" fiili gibi eklerle genişleyen fiillerin veya isimlerin doğru bir şekilde ayrıştırılması gerekir. Bir metinde, "yapılacak" veya "yapılmış" gibi kelimelerin "yapmak" fiili ile olan ilişkisi doğru bir şekilde tespit edilmediğinde, anlamlı kelime öbekleri çıkarılamayabilir. Aynı zamanda, kelimelerin öbek içinde birlikte

kullanım sıklığı, semantik bağlamı ile birlikte değerlendirilmelidir. Bu da, kelime öbeklerinin çıkarımını sadece frekansa dayalı olmaktan çıkarıp, daha derinlemesine dilsel analizlerle destekler. Böylece, metnin gerçek anlamını yansıtan daha doğru ve anlamlı sonuçlar elde edilebilir (Şekil 3).

Sonuç olarak, RAKE yönteminin Türkçe'de etkin bir şekilde kullanılabilmesi için, dilin yapısal ve morfolojik özellikleri dikkate alınarak özelleştirilmiş bir yaklaşım benimsenmelidir. Türkçe'nin eklemeli yapısına uygun olarak işlevsel kelimelerin filtrelenmesi ve kelimeler arasındaki semantik bağlantıların analiz edilmesi, yöntemin doğruluğunu ve etkinliğini artıracaktır. Bu sayede, Türkçe metinlerde daha anlamlı ve bağlama uygun anahtar kelime öbekleri çıkarılabilir, bu da dil işleme çalışmalarında daha başarılı sonuçlar elde edilmesine katkı sağlayacaktır.



Şekil 3. RAKE Algoritma Adımları

C.BERTurk Yöntemi

BERTurk (Schweter, S. 2020), BERT (Bidirectional Encoder Representations from Transformers) modelinin Türkçe diline uyarlanmış özel bir versiyonudur. BERT, metni iki yönlü olarak işleyerek dilin bağlamsal anlamlarını daha derinlemesine anlamayı hedefleyen güçlü bir dil modelidir. BERT'in bu temel prensibi, hem soldan sağa hem de sağdan sola metni okuyarak, kelimelerin ve cümlelerin birbiriyle olan ilişkilerini daha kapsamlı bir şekilde analiz etmeye olanak tanır. BERTurk ise, bu iki yönlü dil modelini Türkçe'nin karmaşık yapısını ve dilsel özelliklerini öğrenmek amacıyla büyük çaplı Türkçe metin korpusları ile eğitilmiş bir model olarak dikkat çeker. Türkçe'nin eklemeli yapısı, zengin bağlamsal anlamları ve özgün sözdizimsel yapıları, bu tür bir model için özel düzenlemeler gerektirdiğinden BERTurk bu zorlukların üstesinden gelmek için tasarlanmıştır.

Türkçe'nin kendine özgü dil yapıları ve morfolojik özellikleri, BERTurk modelinin eğitilmesi ve uygulanması sürecinde dikkate alınan önemli faktörlerdir. Türkçe, eklemeli bir dil olduğundan, kelime kökleri ve ekler birbirine farklı şekillerde bağlanarak yeni anlamlar üretir. Bu da kelimelerin ve cümle yapıların bağlamını doğru bir şekilde anlamayı zorlaştırabilir. Bu nedenle, BERTurk'un başarılı bir performans sergileyebilmesi için, bu bağlamsal zenginlikleri ve dilbilgisel yapıların dinamiklerini etkin bir şekilde modelleyebilmesi gerekmektedir. BERTurk, Türkçe'nin karmaşık dil yapısını öğrenmek amacıyla geniş çaplı Türkçe metinlerle önceden eğitilmiştir. Bu eğitim süreci, modelin dilin sözdizimsel ve anlamsal özelliklerini daha iyi kavramasını sağlar.

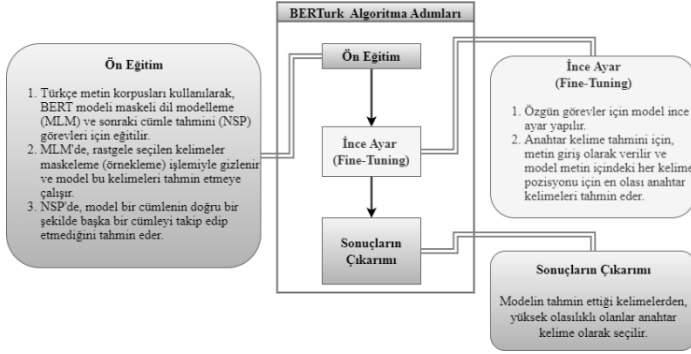
BERTurk, anahtar kelime çıkarımında ve metin işleme görevlerinde başarılı olabilmesi için metin içerisindeki bağlamsal ipuçlarını dikkatle analiz eder. Model, kelimelerin cümle içindeki yerleşimine ve birbirleriyle olan ilişkilerine bakarak, metindeki bağlamı anlamlandırır ve anahtar kelimeleri doğru bir şekilde tahmin etmeye çalışır. Bu bağlamsal analiz süreci, modelin temel işlevlerinden biri olan "masking" yani metin içerisindeki belirli kelimelerin gizlenerek modelin bu boşlukları doldurmasının istenmesiyle gerçekleşir. Bu teknik, BERTurk'un metindeki eksik veya belirsiz kelimeleri tahmin etme yeteneğini geliştirir ve böylece anahtar kelimeleri çıkarırken dilin bağlamını en iyi şekilde kullanmasını sağlar.

Türkçe'nin zengin morfolojik yapısı, BERTurk'un performansını optimize etmek için dikkat edilmesi gereken bir diğer önemli unsurdur. Türkçe'de kelimeler, köklerine eklenen çeşitli eklerle yeni anlamlar kazanır ve bu durum, kelime öbeklerinin doğru bir şekilde analiz edilmesini zorlaştırabilir. BERTurk'un bu zorluğu aşabilmesi için modelin parametreleri, Türkçe'nin morfolojik zenginliğine uygun olarak ayarlanmalıdır. Örneğin, kelime kökleri ve eklerin birbirine olan etkisi, modelin dikkat mekanizmaları tarafından doğru bir şekilde işlenmelidir. Bu da modelin, kelime öbeklerinin anlamlarını ve metin içerisindeki rollerini daha doğru bir şekilde çıkarmasına yardımcı olacaktır.

BERTurk'un performansı, sadece dilin sözdizimsel yapılarının analiz edilmesiyle sınırlı kalmaz; aynı zamanda dilin semantik yapılarının da derinlemesine anlaşılması üzerine kuruludur. Örneğin, model bir metin içerisinde "öğrenci" kelimesini analiz ederken, bu kelimenin hem çoğul hali olan "öğrenciler" hem de diğer

bağlamlarla ilişkili anlamlarını dikkate alarak bir değerlendirme yapar. Bu değerlendirme sürecinde, metindeki diğer kelimelerle olan bağıntılar ve cümle içerisindeki konumlar da göz önünde bulundurularak daha derinlemesine bir bağlamsal anlam çıkarılır.

Sonuç olarak, BERTürk, Türkçe'nin zengin dil yapısını ve morfolojik karmaşıklıklarını anlamlandırmak için optimize edilmiş bir dil modelidir. Anahtar kelime çıkarımı ve metin işleme gibi görevlerde başarılı olabilmesi için, bağlamsal ipuçlarını etkin bir şekilde kullanarak metinlerdeki gizli anlamları doğru bir şekilde analiz eder. Modelin başarısı, Türkçe'nin morfolojik yapısına uygun ayarlamalar ve doğru parametrelerle desteklendiğinde, dil işleme çalışmalarında önemli bir katkı sağlar ve metin analizi süreçlerinde daha güçlü ve verimli sonuçlar elde edilmesine yardımcı olur (Şekil 4).

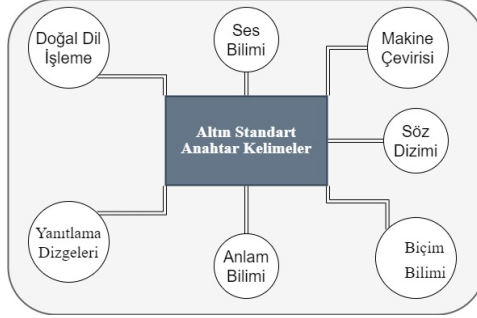


Şekil 4. BERTürk Algoritma Adımları

D.Değerlendirme Metrikleri

Her bir yöntemle çıkarılan anahtar kelimeler, önceden belirlenen altın standart anahtar kelimelerle (Şekil 5) karşılaştırılmıştır. Bu karşılaştırmada, aşağıda detayları verilen

doğruluk, F1 skoru ve geri çağırma oranı gibi metrikler kullanılmıştır. Bu metrikler, her yöntemin ne derece başarılı olduğunu ölçmek amacıyla hesaplanmış ve sonuçlar karşılaştırmalı olarak analiz edilmiştir. Çalışmanın sonunda, hangi yöntemin daha etkili olduğu ve hangi durumlarda hangi yöntemin kullanılmasının daha uygun olacağına dair bulgular elde edilmiştir.



Şekil 5. Altın Standart Anahtar Kelimeler

Doğruluk (Accuracy): Çıkarılan anahtar kelimelerin ne kadarının doğru olduğu hesaplanır.

F1 Skoru: Anahtar kelime çıkarımının doğruluğu ve kapsamını ölçen bir değerlendirme metriğidir.

Recall: Çıkarılan anahtar kelimelerin, tüm doğru anahtar kelimelere kıyasla ne kadarını bulabildiğini ölçen bir metriktir.

Analiz Sonuçları

Bu çalışmada, Türkçe metinlerde anahtar kelime çıkarımı için kullanılan LDA, RAKE ve BERTurk yöntemlerinin performansları özgün bir Türkçe metin kullanılarak analiz edilmiştir. Her bir yöntemin doğruluk, F1 skoru ve geri çağırma oranı gibi metrikler açısından nasıl performans gösterdiği değerlendirilmiştir. Elde

edilen sonuçlar, her bir yöntemin avantajlarını ve sınırlamalarını ortaya koymaktadır.

Elde edilen tüm sonuçlar karşılaştırma tablosunda (Tablo 1) verilmiştir.

Tablo 1: Karşılaştırmalı Analiz Tablosu

Yöntem	Doğruluk (Accuracy)	F1 Skoru	Geri Çağırma (Recall)
LDA	0.75	0.67	0.63
RAKE	0.63	0.58	0.55
BERTurk	0.69	0.63	0.60

Tablo 1’de de görüldüğü üzere çalışmada, Türkçe metinlerde anahtar kelime çıkarımı üzerine LDA, RAKE ve BERTurk yöntemlerinin performansları detaylı bir şekilde karşılaştırılmıştır. Her bir yöntemin kendine özgü güçlü yönleri olduğu ve farklı türdeki metinler için çeşitli avantajlar sunduğu gözlemlenmiştir.

LDA, belgelerdeki genel temaları ve önemli kavramları yakalamada en etkili yöntem olarak öne çıkmıştır. Özellikle geniş kapsamlı metinlerde, metnin altında yatan ana konuları tespit etmek ve bu konuların metin içindeki dağılımını anlamak açısından LDA oldukça başarılı sonuçlar vermiştir. Bu modelin, her bir belgenin birden fazla konudan oluşabileceği varsayımı sayesinde, metinlerin yapısal analizinde önemli bir rol oynadığı görülmüştür.

RAKE yöntemi ise, daha ayrıntılı ve spesifik kelime öbeklerini tespit etme konusunda öne çıkmıştır. RAKE, özellikle metin içindeki anahtar kelimelerin sıklığına ve bunların birlikte kullanımına odaklanarak, metnin ince detaylarını çıkarma yeteneği göstermiştir. Bu yöntem, kelime öbeklerini ve metin içindeki önemli ifadeleri hızlı bir şekilde tespit edebilmesiyle, metnin içerdiği daha küçük

fakat kritik öneme sahip bilgileri yakalamada etkili olmuştur. Türkçe'nin eklemeli yapısına uygun olarak yapılan düzenlemelerle RAKE'in bu alanda verimli sonuçlar sunduğu gözlemlenmiştir.

BERTurk, bağlamsal anlam çıkarma yeteneği ile dikkate değer bir performans sergilemiştir. İki yönlü dil işleme özelliği sayesinde, kelimeleri yalnızca frekanslarına göre değil, aynı zamanda bağlamlarına göre de değerlendirebilmiş ve bu sayede daha derin anlamlar çıkarmıştır. BERTurk, kelimelerin metin içinde hangi anlamda kullanıldığını doğru bir şekilde çözümleyebilme kapasitesi ile dikkat çekmiştir. Ancak, BERTurk'un zaman zaman daha genel ve bağlamsal açıdan çok geniş kelimeler sunması, metnin spesifik detaylarına odaklanmak isteyen çalışmalar için bazı sınırlamalar ortaya koymuştur. Bununla birlikte, bağlamsal anlam çıkarımında güçlü olması, özellikle daha karmaşık ve uzun metinlerde büyük bir avantaj sağlamıştır.

Bu bulgular, Türkçe metinlerde anahtar kelime çıkarımı yaparken hangi yöntemin hangi senaryolar için daha uygun olduğunu anlamak isteyen araştırmacılara rehberlik sunmaktadır. LDA, geniş kapsamlı tematik analizler için ideal bir araçken, RAKE daha detaylı ve spesifik anahtar kelime öbeklerinin çıkarımında etkili bir yöntemdir. BERTurk ise, kelimelerin bağlamsal anlamını çözümlmek ve derinlemesine anlamak isteyen araştırmalar için önemli bir kaynak olarak değerlendirilebilir. Araştırmacılar, metnin yapısına ve ihtiyaçlarına göre bu yöntemlerden birini veya bir kombinasyonunu kullanarak daha doğru ve kapsamlı anahtar kelime çıkarımı yapabilirler.

Buna göre bu çalışma, Türkçe metinlerde farklı anahtar kelime çıkarım yöntemlerinin hangi durumlarda daha verimli olduğunu göstermekte ve her bir yöntemin kendi avantajlarını belirginleştirmektedir. Bu da, metin analizi süreçlerinde doğru yöntemi seçerek, dil işleme çalışmalarını daha verimli ve anlamlı kılmak isteyen araştırmacılara önemli bir rehber sunmaktadır.

Sonuç

Bu çalışmada, Türkçe metinlerde anahtar kelime çıkarımı için kullanılan üç yaygın yöntem olan LDA, RAKE ve BERTurk'un performansları karşılaştırmalı olarak incelenmiştir. Elde edilen sonuçlar, her bir yöntemin farklı metin türlerinde ve koşullarda nasıl performans gösterdiğini ortaya koymuştur. Genel olarak, LDA yöntemi, metnin genel temasını ve önemli kavramlarını belirlemede en başarılı yöntem olarak öne çıkmıştır. RAKE yöntemi, metin içerisindeki önemli kelime öbeklerini tespit etme konusunda güçlü bir performans sergilemiş, ancak daha az spesifik kelimeler üretmiştir. BERTurk yöntemi ise, bağlamsal anlam çıkarımı yaparak doğru anahtar kelimeleri tespit edebilmiş, ancak bazı durumlarda daha genel terimler sunmuştur.

Bu bulgular, Türkçe metinlerde anahtar kelime çıkarımı yaparken kullanılacak en uygun yöntemin seçimi konusunda önemli ipuçları sunmaktadır. Her bir yöntemin avantajları ve sınırlamaları göz önüne alındığında, belirli bir metin türü veya analiz amacına bağlı olarak uygun yöntemin seçilmesi önemlidir. Gelecekteki çalışmalar, bu yöntemlerin daha geniş kapsamlı veri setleri üzerinde test edilmesiyle daha iyi performans gösterebilecek hibrit yaklaşımlar geliştirilmesine katkı sağlayabilir. Bu şekilde, Türkçe metinlerde daha hassas ve etkili anahtar kelime çıkarımı sağlanabilir.

Kaynaklar

Nadim, M., Akopian, D., & Matamoros, A. (2023). A Comparative Assessment of Unsupervised Keyword Extraction Tools. *IEEE Access*.

Ayan, E. T., Arslan, R., Zengin, M. S., Duru, H. A., Salman, S., & Bardak, B. (2021, June). Turkish keyphrase extraction from web pages with BERT. In *2021 29th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.

Tohma, K., & Kutlu, Y. (2020). Challenges Encountered in Turkish Natural Language Processing Studies. *Natural and Engineering Sciences*, 5(3), 204-211.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A. M., Nunes, C., & Jatowt, A. (2018, March). A text feature based automatic keyword extraction method for single documents. In *European conference on information retrieval* (pp. 684-691). Cham: Springer International Publishing.

Erdađı, E., & Tunali, V. (2024). Comparison of feature-based sentence ranking methods for extractive summarization of turkish news texts. *Sigma Journal of Engineering and Natural Sciences*, 42(2), 321-334.

Krivorogov, D., Nizamov, T., Fazlyev, A., Hodyrev, A., Shusharin, D., & Glazkova, A. (2023). Developing a Service for Collecting and Analyzing Electives Reviews. *Vestnik NSU. Series: Information Technologies*.

Issa, B., Jasser, M. B., Chua, H. N., & Hamzah, M. (2023, October). A comparative study on embedding models for keyword

extraction using KeyBERT method. In *2023 IEEE 13th International Conference on System Engineering and Technology (ICSET)* (pp. 40-45). IEEE.

Schweter, S. (2020). BERTurk-BERT models for Turkish. *Zenodo*, 2020, 3770924.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan), 993-1022.

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1-20.

Göz, F., Mutlu, A., Küçük, K., Temur, M., & Gün, A. (2021, June). Effect of Centrality Measures for Keyword Extraction from Turkish Documents. In *2021 29th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.

Eryiğit, G., Adali, K., Torunoğlu-Selamet, D., Sulubacak, U., & Pamay, T. (2015, June). Annotation and extraction of multiword expressions in Turkish treebanks. In *Proceedings of the 11th workshop on multiword expressions* (pp. 70-76).

Ozdemir, B., & Cicekli, I. (2009, September). Turkish keyphrase extraction using multi-criterion ranking. In *2009 24th International Symposium on Computer and Information Sciences* (pp. 269-273). IEEE.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2020). YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509, 257-289.

Mihalcea, R., & Tarau, P. (2004, July). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 404-411).

BÖLÜM IV

Türk Dilleri İçin Doğal Dil İşleme: Zorluklar, Ortak Özellikler ve Çözümler

Halil İbrahim OKUR¹
Kadir TOHMA²

Giriş

Türk dilleri, eklemeli yapıları, zengin morfolojik özellikleri ve geniş coğrafi dağılımları nedeniyle doğal dil işleme (DDİ) alanında önemli bir araştırma konusu olarak öne çıkmaktadır (Kornfilt, J. 2018). Bu diller, tarihsel ve kültürel bağlamlarda geniş bir coğrafi bölgede konuşulmakta olup, Türk dili ailesinin üyeleri arasında Türkçe, Azerice, Özbekçe, Kazakça, Kırgızca, Türkmence ve birçok diğer dil bulunmaktadır. Bu dillerin ortak morfolojik ve sentaktik

¹ Dr., İskenderun Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü, İskenderun/Türkiye, Orcid: 0000-0003-0339-4626, hibrahim.okur@iste.edu.tr

² Dr. Öğr. Üyesi, İskenderun Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü, İskenderun/Türkiye, Orcid: 0000-0002-2631-7810, kadir.tohma@iste.edu.tr

özellikleri, dilbilimi ve bilgisayarlı dil işleme alanında benzersiz bir araştırma alanı oluşturmaktadır (Kulamshaev, K., 2019).

Türk dilleri, yeni kelimelerin köklere çeşitli eklerin eklenmesiyle oluşturulduğu eklemeli yapılarıyla karakterize edilir. Bu durum morfolojik zenginliği artırmakla birlikte, kelime kökleri ve eklerin doğru analizini zorlaştırır. Örneğin, Türkçede "ev" kelimesi, çoğul eki "ler" eklenerek "evler" olur. Benzer şekilde, Özbekçede "kitob" (kitap) kelimesi, çoğul eki "lar" eklenerek "kitoblar" (kitaplar) haline gelir. Bu eklemeli yapıların analizi, DDİ uygulamaları için özel algoritmalar ve yöntemler gerektirir.

Türk dillerinin geniş coğrafi dağılımı, DDİ çalışmalarında çeşitli lehçelerin ve varyantların dahil edilmesini zorunlu kılar. Bu diller, Türkiye, Azerbaycan, Özbekistan, Kazakistan, Kırgızistan, Türkmenistan ve Çin'in Sincan Uygur Özerk Bölgesi gibi bölgelerde konuşulmaktadır. Bu bölgelerde konuşulan lehçeler fonetik, morfolojik ve sentaktik farklılıklar göstermekte olup, dil modelleri ve DDİ algoritmaları geliştirirken dikkate alınması gereken önemli bir faktördür (Ofłazer, K., 2016).

Türk dilleri üzerine DDİ alanında yapılan çalışmalar çeşitli zorluklar sunmaktadır. Bu dillerin eklemeli yapısı, kelime formlarının doğru analiz edilmesini gerektirir (Ofłazer, K., 2016). Morfolojik analiz, kelime köklerinin ve eklerin doğru bir şekilde tespit edilmesini içerir ve dil modellerinin doğruluğunu artırmak açısından büyük önem taşır. Ayrıca, bazı Türk dillerinin yeterli veri ve kaynak eksikliği nedeniyle düşük kaynaklı diller olarak sınıflandırılması, DDİ çalışmalarında önemli bir engel

oluşturmaktadır. Bu sorun, özellikle Kırgızca, Çuvaşça ve Saha (Yakut) gibi diller için belirgindir.

Alfabe ve karakter sorunları da Türk dilleri üzerine yapılan DDİ çalışmalarında önemli zorluklar sunmaktadır (Tohma, K., et al., 2020; Safaya, A., et al., 2022). Farklı Türk dilleri farklı alfabeler kullanmakta olup, bazı özel karakterler ASCII standartlarına uymayabilir. Örneğin, Uygurca Arap alfabesini kullanırken, Tatarca Kiril alfabesini kullanır. Bu durum, dil verilerinin standardizasyonunu ve analizini zorlaştırmaktadır. Ayrıca, kod değiştirme ve çok dillilik gibi dilsel olgular Türk dillerinde yaygındır ve DDİ uygulamalarının karmaşıklığını artırmaktadır. Örneğin, bir Azerice konuşan kişi aynı cümlede hem Azerice hem de Rusça kelimeler kullanabilir, bu da dil modellerinin karmaşıklığını artırmaktadır.

Makine çevirisi, Türk dilleri için DDİ alanında önemli bir alt disiplindir. Türk dillerinin eklemeli yapısı, kelimelerin doğru çevirisini zorlaştırmaktadır. Örneğin, Türkçe "iş" kelimesi, çeşitli eklerle birleştirilerek "işçi", "işler" ve "işsizlik" gibi çeşitli kelimeler oluşturulabilir. Aynı olgu diğer Türk dillerinde de gözlemlenir (Johanson, L., 2021). Bu nedenle, kelime köklerinin ve eklerin doğru bir şekilde tespit edilmesi, makine çeviri sistemlerinin doğruluğunu artırmak açısından kritik öneme sahiptir. Bu alanda Türk dilleri için birçok çalışma yapılmıştır (Mirzakhlov, J., et al., 2021; Alkım, E., et al., 2019; Tantuğ, A. C., et al., 2018; Mirzakhlov, J., 2021).

Morfolojik analiz, Türk dilleri üzerine yapılan DDİ çalışmalarında önemli bir rol oynamaktadır. Bu analiz, kelime köklerinin ve eklerin doğru bir şekilde tespit edilmesine yardımcı

olarak dil modellerinin doğruluğunu artırmaktadır (Tukeyev, U., et al., 2021).

Türk dilleri üzerine yapılan DDİ çalışmalarında karşılaşılan bu zorluklara çözüm bulmak için çeşitli yaklaşımlar ve yöntemler geliştirilmiştir. Morfolojik analiz, kelime köklerini ve ekleri doğru bir şekilde tespit etmek için kullanılan temel yöntemlerden biridir. Ayrıca, çok dilli modellerin geliştirilmesi, Türk dilleri arasındaki benzerliklerden yararlanarak performansı artırabilir. Örneğin, Turk Morphology Portal, bu zorlukları ele almak ve Türk dilleri için dilbilimsel yazılım geliştirilmesini desteklemek amacıyla geliştirilmiştir. Portal, eklemeli yapıların analiz edilmesi ve çok dilli modellerin geliştirilmesi için değerli bir kaynak sunmaktadır (Gatiatullin et al., 2020). Diğer araçlar, morfoAnalyse (Abdurakhmonova, N., et al., 2022), UzMorphoAnalyzer (Mengliev, D., et al., 2021) ve diğerleri (ITU NLP 2024, Abdurakhmonova, N. Z. et al., 2022, Zhanabergenova, D., et al., 2021, Tukeyev, U., 2023, Bekeyeva, N., et al., 2021, Mussakhojayeva, S., et al., 2023), Türk dillerinin morfolojik analizine önemli katkılarda bulunmuştur.

Türk dilleri üzerine yapılan DDİ çalışmalarında bu zorlukların üstesinden gelmek için çeşitli yöntemler ve yaklaşımlar geliştirilmiştir. Gelişmiş morfolojik analiz ve anlamsal ayrıştırma yöntemleri, dil modellerinin doğruluğunu artırmak için kritik öneme sahiptir. Bu bağlamda, dilsel çalışmalar ve transfer öğrenme teknikleri, düşük kaynaklı dillerde DDİ performansını artırmak için etkili stratejilerdir. Ayrıca, Türk dilleri arasındaki fonetik benzerlikler ve ortak kelime yapıları, bu diller arasında doğru bağlantılar kuran dil modellerinin geliştirilmesini kolaylaştırabilir ve DDİ uygulamalarını daha etkili hale getirebilir. Bu çabalar, dil

modellerinin doğruluğunu artırmak ve daha kapsamlı ve esnek DDİ uygulamaları geliştirmek için önemli bir temel oluşturmaktadır. Türk dilleri için geliştirilen yenilikçi yöntemler ve araçlar, gelecekteki DDİ çalışmalarının başarısı ve verimliliği açısından önemli adımlar teşkil etmektedir.

Bu makale, Türk dilleri için DDİ alanında yapılan çalışmaların zorluklarını, ortak özelliklerini ve örneklerini kapsamlı bir şekilde gözden geçirmektedir. Türk dilleri üzerine yapılan DDİ çalışmalarının mevcut durumunu ve gelecekteki yönelimlerini değerlendirmeyi, bu alandaki anlayışı artırmayı ve daha etkili çözümler geliştirmeyi amaçlamaktadır.

Türk Dillerinin Ortak Özellikleri ve DDİ Çalışmalarındaki Zorluklar

Türk dillerinin yapısal özellikleri, DDİ çalışmalarında çeşitli zorluklar ortaya çıkarmaktadır. Bu zorluklar, dilin morfolojik zenginliğinden, düşük kaynaklı dillerin varlığından, alfabe ve karakter sorunlarından, çok dillilikten ve kod değiştirme olgusundan kaynaklanmaktadır. Bu bölümde, Türk dilleri üzerindeki DDİ çalışmalarında karşılaşılan bu zorluklar ayrıntılı olarak ele alınmaktadır.

Benzer Özelliklere Sahip Diller

Türk dillerinin büyük bir kısmı dilbilimsel ve morfolojik özelliklerde büyük benzerlikler sergilemektedir. Bu benzerlikler, dilbilimsel araştırmalar ve DDİ çalışmalarında ortak yöntemlerin kullanılmasına olanak tanımaktadır. Ortak benzerlikler gösteren bazı Türk dilleri şunlardır:

- **Türkiye Türkçesi:** Türkçe, en yaygın konuşulan Türk dili olup, yapısal özellikleri ve kelime türetme yöntemleri diğer Türk dilleri ile büyük benzerlik göstermektedir.
- Örnek: "Gel" → "Geldi" → "Gelecek"
- **Azerbaycan Türkçesi:** Azerbaycan'da konuşulan Azerbaycan Türkçesi, Türkçe ile yüksek derecede benzerlik taşır ve birçok ortak kelime ve dil bilgisel kuralları paylaşır.
- Örnek: "Gəl" → "Gəldi" → "Gələcək"
- **Türkmen Türkçesi:** Türkmenistan'da konuşulan Türkmen Türkçesi, özellikle eklemeli yapısı ile Türkçe ve Azerbaycan Türkçesi ile yapısal benzerlikler taşır.
- Örnek: "Gel" → "Geldi" → "Gelejek"
- **Özbek Türkçesi:** Özbekistan'da konuşulan Özbek Türkçesi, diğer Türk dilleriyle benzer bir eklemeli yapı ve morfolojik özellikler taşımakla birlikte, Arapça ve Farsçadan eklenen etkiler de mevcuttur.
- Örnek: "Kel" → "Keldi" → "kelajak "

- **Kazak Türkçesi:** Kazakistan'da konuşulan Kazak Türkçesi, hem Latin hem de Kiril alfabesini kullanır ve zengin morfolojik yapısıyla bilinir.

Örnek: "Kel (кел)" → "Keldi (келді)" → "keleşek (келешек)"

- **Kırgız Türkçesi:** Kırgızistan'da konuşulan Kırgız Türkçesi, benzer eklemeli yapıya sahiptir ve genellikle Kiril alfabesiyle yazılmaktadır.
- Örnek: "Kel" → "Keldi" → "Keleşek"
- **Uygur Türkçesi:** Çin'in Sincan Uygur Özerk Bölgesi'nde konuşulan Uygur Türkçesi, Türkçe ve diğer Türk dilleriyle yapısal benzerlikler taşır. Geleneksel olarak Arap alfabesiyle yazılmakta olup, geçmişte Kiril ve Latin alfabeleri de kullanılmıştır ve bazı bağlamlarda halen kullanılmaktadır.
- Örnek: "Kel" → "Keldi" → "Kelidu"
- **Tatar Türkçesi:** Tataristan ve çevresinde konuşulan Tatar Türkçesi, eklemeli yapısı ve zengin morfolojisi ile bilinir. Sovyet döneminde Tatar Türkçesi Kiril alfabesiyle yazılmaya başlanmış ve günümüzde de kullanılmaktadır. Bununla birlikte, bazı Tatar toplulukları Latin alfabesini de kullanmaktadır.
- Örnek: "Kil" → "Kildi" → "Kiläçäk"

Bu dillerin ortak özellikleri, kelime köklerine ekler eklenerek yeni kelimeler oluşturulan eklemeli yapıların kullanımı ve zengin morfolojik özelliklerdir. Bu özellikler, dilbilimsel arařtırmalar ve doęal dil iřleme alıřmalarında önemli fırsatlar ve zorluklar sunmaktadır. Her bir dilin kendine özgü alfabeleri ve bölgesel leheleri, dijital iřlemeyi daha karmařık hale getirmekte ve özel analiz yöntemleri gerektirmektedir. Bu nedenle, bu diller üzerinde yapılan alıřmalar, dil modellerinin doęruluęunu artırmak ve dillerin tüm eřitlilięini kapsamak için titiz veri toplama ve analiz süreçlerini zorunlu kılmaktadır.

Daha Az Benzerlik Gösteren veya Ortak Özellik Göstermeyen Diller

Bazı Türk dilleri, coęrafi uzaklık, tarihsel gelişim ve kültürel etkileşimler nedeniyle dięer Türk dilleriyle daha az benzerlik göstermektedir. Bu diller, dilbilimsel arařtırmalarda ve DDI alıřmalarında daha dikkatli bir şekilde ele alınmayı gerektirebilir. Daha az ortaklık veya hibir benzerlik göstermeyen bazı Türk dilleri řunlardır:

• **Saha (Yakut) Türkesi:** Sibirya'da konuşulan bu dil, dięer Türk dillerinden önemli ölçüde farklı yapısal özelliklere sahiptir.

• **uvař Türkesi:** uvařistan' da konuşulan bu dil, dil bilgisel ve morfolojik özellikleri bakımından dięer Türk dillerine göre daha farklıdır.

• **Hala Türkesi:** İran'da konuşulan bu dil, dięer Türk dillerinden belirgin şekilde farklı bir dil bilgisel yapıya sahiptir.

Tablo 2. Ortak Benzerlikler Gösteren Dillere ve Daha Az Benzerlik veya Ortaklık Göstermeyen Dillere Örnekler

Dil	Bölge	Alfabe Kullanımı	Yapısal Özellikler	Örnek (gel, geldi, gelecek)
Türkçe	Türkiye	Latin	Eklemeli yapı, zengin morfoloji	Gel, Geldi, Gelecek
Azerbaycan Türkçesi	Azerbaycan	Latin	Türkçeye yüksek benzerlik	Gəl, Gəldi, Gələcək
Türkmen Türkçesi	Türkmenistan	Latin	Türkçe ve Azerbaycan Türkçesine benzer	Gel, Geldi, Gelejek
Özbek Türkçesi	Özbekistan	Latin, Kiril	Eklemeli yapı, Arapça ve Farsça etkisi	Kel, Keldi, Kelajak
Kazak Türkçesi	Kazakistan	Latin, Kiril	Zengin morfolojik yapı	Kel (кел), Keldi (келді), Keleşek (келешек)
Kırgız Türkçesi	Kırgızistan	Kiril	Eklemeli yapı	Kel, Keldi, Keleşek
Uygur Türkçesi	Çin (Sincan Uygur Özerk Bölgesi)	Arap, Kiril, Latin	Türkçeye benzer, tarihsel alfabe kullanımı	Kel, Keldi, Kelidu
Tatar Türkçesi	Rusya (Tataristan ve çevresi)	Kiril, Latin	Eklemeli yapı, zengin morfoloji	Kil, Kildi, Kiläçäk
Saha (Yakut) Türkçesi	Sibirya	Kiril	Diğer Türk dillerinden farklı yapısal özellikler	-
Çuvaş Türkçesi	Çuvaşistan	Kiril	Farklı dilbilgisel ve morfolojik özellikler	-
Halaç Türkçesi	İran	Latin	Farklı dilbilgisel yapı	-

Bu dillerdeki farklılıklar, coğrafi uzaklık, tarihsel gelişim ve diğer dillerle olan etkileşimlerden kaynaklanmaktadır. Bu nedenle, bu dilleri kapsayan DDİ çalışmalarında özel yöntemler ve yaklaşımlar gereklidir. Tablo 1'de, Ortak Benzerlik Gösteren Diller

ve Daha Az Benzerlik Gösteren veya Ortak Özellik Göstermeyen Diller için örnekler sunulmuştur.

Morfolojik Zenginlik ve Eklemeli Yapı

Türk dillerinin eklemeli yapısı, kelime köklerine ekler eklenerek yeni kelimelerin türetilmesine olanak tanımaktadır. Bu özellik, morfolojik analizi ve kelime köklerinin doğru bir şekilde tanımlanmasını zorlaştırmaktadır. Örneğin, Kazakçada "kitap" (kitap) kelimesi, farklı ekler eklenerek "kitaptar" (kitaplar) veya "kitaptan" (kitaptan) şeklinde dönüştürülebilir. Benzer şekilde, Türkçede "çalış" (çalışmak) kelimesi, "çalışıyor" (çalışıyor) ve "çalıştı" (çalıştı) gibi fiil formlarına dönüştürülebilir. Bu morfolojik süreçler, kelime yapılarını analiz eden DDİ algoritmalarının karmaşıklığını artırmaktadır.

Türk dillerinin morfolojik zenginliği, kelime köklerinin ve eklerin hassas bir şekilde tanımlanmasını gerektirmektedir. Örneğin, Kırgızcada "iş" (çalışma) kelimesi, "iştep" (çalışmalar) ve "işten" (işten) formlarına dönüştürülebilir. Bu morfolojik yapıların doğru analizi, DDİ uygulamalarının doğruluğunu ve verimliliğini artırmaktadır. Ayrıca, kelime köklerinin ve eklerin doğru analizi, dilin esnekliğini ve ifade gücünü artırırken, morfolojik analiz süreçlerini daha karmaşık hale getirmektedir.

Düşük Kaynaklı Diller ve Veri Eksikliği

Birçok Türk dili için yeterli dil verisi ve kaynaklarının bulunmaması, DDİ çalışmalarında önemli bir zorluk oluşturmaktadır. Bu sorun, özellikle Kırgızca, Çuvaşça ve Saha (Yakut) gibi dillerde belirgindir. Düşük kaynaklı diller, dil modellerinin eğitilmesi ve test edilmesi için yeterli veri

sağlamadığından, bu modellerin doğruluğu ve genel performansı olumsuz etkilenmektedir (Magueresse ve ark., 2020; Ranathunga ve ark., 2023).

Düşük kaynaklı diller için daha fazla dil verisi toplanması ve paylaşılması, DDİ çalışmalarının başarısını artırabilir. Örneğin, Kırgızca için dil verilerinin artırılması ve bu verilerin paylaşılması, dil modellerinin doğruluğunu artırabilir. Benzer çabalar Çuvaşça ve Saha (Yakut) için de gösterilmelidir. Veri toplama ve paylaşma, bu dillerin DDİ alanında kullanımını genişletebilir.

Veri eksikliğinin giderilmesi için veri çeşitlendirme ve zenginleştirme, çapraz dilli çalışmalar ve transfer öğrenimi, otomatik etiketleme ve doğal dil anlama, konuşma işleme ve konuşma tanıma, açık kaynaklı araçların desteklenmesi ve topluluk katkılarının teşvik edilmesi gibi stratejiler kritik önem taşımaktadır. Farklı bölgelerde konuşulan lehçeler, günlük konuşmalar, edebi eserler ve dijital metinler gibi çeşitli kaynaklardan veri toplamak, daha geniş ve çeşitli veri kümeleri oluşturabilir (Madaan ve ark., 2020). Yüksek kaynaklı Türk dillerinden düşük kaynaklı dillere bilgi transferi, DDİ performansını iyileştirebilir (Ranathunga ve ark., 2023). Otomatik etiketleme ve doğal dil anlama tekniklerinin geliştirilmesi, metin analizlerinin daha doğru yapılmasını sağlayabilir (Magueresse ve ark., 2020). Konuşma işleme ve tanıma sistemleri, düşük kaynaklı dillerin dijital asistanlar, çağrı merkezi uygulamaları ve dil öğrenme araçlarında kullanılmasını kolaylaştırabilir. Açık kaynak projelerinin desteklenmesi ve topluluk katkılarının teşvik edilmesi, dil modelleri ve analiz araçlarının geliştirilmesini hızlandırabilir (Ranathunga ve ark., 2023).

Fonetik Benzerlikler

Türk dilleri arasında fonetik benzerlikler bulunmaktadır (Kornfilt, 2018), bu da kelime kökleri ve eklerin benzer olmasına yol açmaktadır. Ancak, bu benzerlikler homograf (yazılışı aynı fakat anlamları farklı olan kelimeler) varlığı nedeniyle bazen zorluklara sebep olabilir. Örneğin, Türkçede "yüz" kelimesi hem "yüzmek" hem de "yüz (insan yüzü)" anlamına gelebilir. Bu tür homograflar, dil modellerinin bağlamı doğru bir şekilde analiz etmesini zorlaştırır.

Fonetik benzerlikler, Türk dilleri arasındaki yakın ilişkiyi de gösterir. Örneğin, Azerbaycanca "ev" kelimesi (ev) ile Türkçedeki "ev" kelimesi aynı anlama gelir. Benzer şekilde, Uygurca "kitap" kelimesi (kitap), Türkçedeki "kitap" ile aynıdır. Bu tür benzerlikler, dil modellerinin bu diller arasında geçiş yapmasını kolaylaştırabilir, ancak yanlış yorumlama riskini de beraberinde getirir.

Fonetik benzerliklerin yanı sıra, Türk dillerinin ortak kelime yapıları da dikkate alınmalıdır. Örneğin, Kazakçadaki "mektep" (okul), Kırgızca "mektep" ve Özbekçe "maktab" kelimeleri aynı kökten türetilmiş olup aynı anlama gelmektedir. Bu tür benzerlikler, dil modellerinin bu diller arasında doğru bağlantılar kurmasını kolaylaştırabilir, ancak aynı zamanda yanlış yorumlama riskini de artırabilir.

Alfabe ve Karakter Sorunları

Türk dillerinde farklı alfabelerin ve ASCII standartlarına uymayan özel karakterlerin kullanılması, DDİ araçlarının geliştirilmesinde zorluklar yaratmaktadır. Örneğin, Uygurca Arap alfabesini kullanırken, Tatarca Kiril alfabesini kullanmaktadır. Bu durum, dil verilerinin standardizasyonunu ve analizini zorlaştırır.

Ayrıca, ASCII standartlarına uymayan bazı özel karakterler, bu karakterlerin doğru işlenmesini zorlaştırmaktadır.

Alfabe ve karakter sorunları, dil modellerinin doğruluğunu ve performansını olumsuz yönde etkilemektedir. Örneğin, Uygurca metinlerin doğru analizi, Arap alfabesi karakterlerinin doğru bir şekilde tanımlanmasını ve işlenmesini gerektirir. Benzer şekilde, Tatarca metinlerin doğru analiz edilmesi için Kiril alfabesi karakterlerinin doğru bir şekilde tanımlanması ve işlenmesi gerekmektedir. Bu alfabe ve karakter sorunları, dil modellerinin geliştirilmesini ve uygulanmasını zorlaştırmaktadır.

Çok Dillilik ve Kod Değiştirme

Türk dillerinde çok dillilik ve kod değiştirme (aynı konuşmada birden fazla dili kullanma) yaygın olarak görülmektedir. Bu durum, özellikle konuşma tanıma ve çeviri sistemlerinde ek zorluklar yaratmaktadır (Mirzakhlov ve ark., 2021). Örneğin, bir Azerbaycanlı konuşmacı aynı cümlede hem Azerice hem de Rusça kelimeler kullanabilir. Bu tür kod değiştirme durumları, dil modellerinin anlamların bağlamını doğru bir şekilde analiz etmesini zorlaştırarak yanlış yorumlamalara yol açmaktadır.

Kod değiştirme, dil modellerinin karmaşıklığını artırmakta ve DDİ uygulamalarının doğruluğunu olumsuz etkilemektedir. Örneğin, aynı cümlede Azerice ve Rusça kelimelerin kullanılması, dil modellerinin bu kelimeler arasındaki bağlamı doğru anlamasını zorlaştırmaktadır. Bu durum, çeviri sistemlerinin ve konuşma tanıma sistemlerinin doğruluğunu ve performansını olumsuz yönde etkilemektedir.

DDİ Çalışmaları İçin Çıkarımlar

Türk dillerindeki zorluklar, DDİ çalışmalarında çeşitli yöntemler ve yaklaşımlar kullanılarak aşılabilir. Morfolojik analiz, kelime köklerinin ve eklerin doğru bir şekilde tanımlanmasını sağlar ve dil modellerinin doğruluğunu artırır. Ayrıca, çok dilli modeller, Türk dilleri arasındaki benzerliklerden yararlanarak performansı artırabilir.

Gelecekte, düşük kaynaklı Türk dilleri için daha fazla veri toplanması ve paylaşılması, DDİ çalışmalarının başarısını artıracaktır. Ayrıca, Türk dillerinin morfolojik zenginliğini göz önünde bulundurarak morfolojik analiz ve anlamsal belirsizliği giderme yöntemlerinin geliştirilmesi, daha gelişmiş DDİ uygulamalarının geliştirilmesine olanak tanıyacaktır.

Gelecek Yönelimler ve Öneriler

Türk dillerinde yapılacak DDİ çalışmalarına yönelik gelecek yönelimler ve öneriler şunları içermektedir:

- **Veri Toplama ve Paylaşma:** Düşük kaynaklı Türk dilleri için veri toplama ve paylaşımının artırılması, DDİ çalışmalarının başarısını artıracaktır.
- **Çok Dilli Modellerin Geliştirilmesi:** Türk dilleri arasındaki benzerliklerden yararlanarak çok dilli DDİ modellerinin geliştirilmesi, düşük kaynaklı dillerin performansını iyileştirebilir.
- **Morfolojik Analiz ve Anlamsal Belirsizliği Giderme:** Türk dillerinin morfolojik zenginliğini göz önünde bulundurarak, gelişmiş morfolojik analiz ve

anlamsal belirsizliđi giderme yöntemleri geliştirilmelidir.

- **Kod Deđiştirme ve Çok Dillilik:** Kod deđiştirme ve çok dillilik üzerine daha fazla araştırma yapılmalı ve bu olgulara yönelik DDİ araçları geliştirilmelidir.

Ek Araştırma Alanları ve Geliştirme Önerileri

Türk dilleri üzerine yapılan doğal dil işleme (DDİ) çalışmalarının başarısı, yeterli ve çeşitli dil verilerinin kullanılmasına büyük ölçüde bađlıdır. Bu nedenle, veri toplama süreçlerinde daha fazla çeşitlilik sağlanması gerekmektedir. Farklı bölgelerde konuşulan lehçeler, günlük dil, edebi eserler ve dijital metinler, dil modellerinin eğitilmesinde önemli kaynaklar olarak kullanılabilir. Bunun yanı sıra, sosyal medya gönderileri gibi kullanıcı tarafından üretilen içerikler, dil modellerinin daha geniş ve çeşitli veri setleri ile beslenmesine katkı sağlayarak DDİ çalışmalarını zenginleştirebilir (Madaan ve ark., 2020). Veri çeşitliliđi, bu çalışmaların kapsamını genişletirken, düşük kaynaklı Türk dillerine yönelik yapılan çalışmalarda da kritik bir rol oynamaktadır.

Türk dilleri arasında belirli benzerlikler olmasına rağmen, bazı dillerin düşük kaynaklı olması, bu diller için veri yetersizliđi sorununu beraberinde getirmektedir. Bu durum, transfer öğrenimi tekniklerinin kullanımını zorunlu kılmaktadır. Yüksek kaynaklı Türk dillerinden elde edilen bilgi, düşük kaynaklı dillere aktarılabilir ve böylece DDİ performansı önemli ölçüde artırılabilir. Örneđin, Türkçe ve Azerbaycan Türkçesi arasındaki benzerlikler, Özbekçe veya Kırgızca gibi daha az veri bulunan dillere yönelik çalışmalarda

fayda sağlayabilir. Bu sayede, Türk dilleri arasındaki benzerliklerden yararlanarak, DDİ çalışmalarında daha iyi sonuçlar elde edilebilir.

Bu süreçte, veri çeşitliliğinin yanı sıra, Türk dilleri üzerine yapılan DDİ çalışmalarında otomatik etiketleme ve doğal dil anlama tekniklerinin geliştirilmesi de büyük önem taşımaktadır. Bu teknikler, dil modellerinin metinleri daha doğru bir şekilde analiz etmesine olanak tanır. Makine öğrenimi ve derin öğrenme algoritmalarının etkin bir şekilde kullanılması, kelime öbeklerinin ve cümle yapıların daha iyi anlaşılmasını sağlayarak DDİ süreçlerini iyileştirebilir (Magueresse ve ark., 2020). Özellikle bu tür teknolojiler, düşük kaynaklı dillerde karşılaşılan zorlukların üstesinden gelmede önemli bir rol oynamaktadır.

Dil verilerinin doğru bir şekilde işlenmesi sadece metinlerle sınırlı kalmamalıdır. Türk dilleri üzerine yapılan DDİ çalışmalarında konuşma işleme ve konuşma tanıma sistemleri de kritik bir öneme sahiptir. Bu sistemler, ses tanıma teknolojilerinin geliştirilmesini desteklerken, Türk dillerinin dijital asistanlar, çağrı merkezi uygulamaları ve dil öğrenme araçları gibi alanlarda kullanımını da kolaylaştırmaktadır. Farklı lehçeleri ve aksanları tanıyabilen konuşma tanıma sistemlerinin geliştirilmesi, bu teknolojilerin kullanımını daha da yaygınlaştıracaktır.

Türk dilleri arasındaki fonetik ve morfolojik benzerlikler dikkate alındığında, çok dilli modellerin ve dil çiftlerinin geliştirilmesi de büyük bir önem taşımaktadır. Bu modeller, diller arasındaki benzerliklerden yararlanarak daha etkili dil işleme algoritmaları oluşturur. Özellikle Türkçe-Azerbaycan Türkçesi veya

Türkçe-Özbekçe gibi dil çiftleri kullanılarak, dil modellerinin doğruluğu artırılabilir. Böylece, dil işleme süreçlerinde daha verimli ve başarılı sonuçlar elde edilmesi mümkün olacaktır.

Açık kaynak araçlar ve topluluk katkıları da bu çalışmaların başarısında büyük rol oynamaktadır. Türk dilleri için geliştirilen açık kaynak projelerinin desteklenmesi ve topluluk katkılarının teşvik edilmesi, dil modellerinin ve analiz araçlarının geliştirilme sürecini hızlandıracaktır. Bu süreç, araştırmacılar ve geliştiriciler arasında iş birliğini artırarak Türk dillerinin dijitalleşme sürecine katkıda bulunacak ve daha fazla veri kaynağına erişim sağlayacaktır.

Son olarak, Türk dilleri üzerine yapılan DDİ çalışmalarının uzun vadeli başarısı, bu alandaki eğitim ve farkındalığın artırılmasına bağlıdır. Üniversitelerde ve araştırma kurumlarında DDİ eğitiminin genişletilmesi, bu alanda uzmanlaşmış araştırmacıların ve geliştiricilerin yetişmesini sağlayacaktır. Ayrıca, Türk dillerinin dijitalleşme süreci hakkında farkındalık yaratmak, bu dillerin korunmasına ve gelişimine katkıda bulunacak ve dil teknolojilerinin daha etkin bir şekilde kullanılmasına olanak tanıyacaktır.

Türk dilleri, doğal dil işleme (DDİ) alanında büyük bir potansiyele sahiptir; ancak bu potansiyelin tam anlamıyla gerçekleşmesi, karşılaşılan zorlukların aşılması ve önerilen çözümlerin hayata geçirilmesi ile mümkün olacaktır. Bu makalede tartışılan zorluklar ve çözüm önerileri, Türk dilleri üzerine yapılan DDİ çalışmalarının daha iyi anlaşılmasına ve bu alanda daha etkili çözümler geliştirilmesine katkıda bulunmayı amaçlamaktadır.

Sonuç ve Tartışma

Türk dilleri, doğal dil işleme (DDİ) alanında büyük bir potansiyele sahip olmakla birlikte, çeşitli zorluklar da sunmaktadır. Bu makale, Türk dilleri üzerine yapılan DDİ çalışmalarındaki zorlukları, bu dillerin ortak özelliklerini ve yürütülen çalışmalardan örnekleri ele almaktadır. Türk dillerinin eklemeli yapısı, morfolojik zenginliği ve geniş coğrafi yayılımı, dilbilimsel araştırmalar ve DDİ çalışmaları için önemli fırsatlar sunarken, aynı zamanda çeşitli zorlukları da beraberinde getirmektedir. Morfolojik zenginlik, kelime kökleri ve eklerin doğru bir şekilde tanımlanmasını gerektirirken, düşük kaynaklı dillerin varlığı, yeterli veri ve kaynak sağlanmasını engelleyen önemli bir bariyer oluşturmaktadır. Ayrıca, farklı alfabe ve karakterlerin kullanımı, dil verilerinin standardizasyonunu ve analizini zorlaştırırken, çok dillilik ve kod değiştirme gibi dilsel olgular, dil modellerinin karmaşıklığını artırmaktadır.

Bu zorlukların üstesinden gelmek için, Türk dilleri üzerine yapılan DDİ çalışmalarında gelişmiş morfolojik analiz yöntemleri ve çok dilli modeller kullanılmalıdır. Morfolojik analiz, kelime kökleri ve eklerin doğru bir şekilde tanımlanmasını sağlarken, çok dilli modeller, Türk dilleri arasındaki benzerliklerden yararlanarak performansı iyileştirebilir. Bu çalışmaların yanı sıra, Türk dillerinde çok dillilik ve kod değiştirme olgularının incelenmesi de büyük önem taşımaktadır. Kod değiştirme ve çok dillilik, özellikle konuşma tanıma ve çeviri sistemlerinde ek zorluklar yaratmakta ve dil modellerinin anlamların bağlamını doğru bir şekilde analiz etmesini zorlaştırmaktadır. Bu nedenle, bu alanlardaki gelecekteki

arařtırmalar, Trk dillerinin DDİ alanındaki kullanımını daha da geliřtirecek ve geniřletecektir.

Gelecekte, Trk dilleri zerine yapılan DDİ alıřmalarının daha da geliřmesi beklenmektedir. Dřk kaynaklı Trk dilleri iin daha fazla veri toplanması ve paylařılması, DDİ alıřmalarının bařarısını artıracaktır. zellikle Kırgızca, uvařça ve Saha (Yakut) gibi dillere ynelik dil verilerinin artırılması ve bu verilerin paylařılması, dil modellerinin doęruluęunu ve genel performansını iyileřtirecektir. Ayrıca, Trk dillerinin morfolojik zenginlięi gz nnde bulundurularak morfolojik analiz ve anlamsal belirsizlięi giderme yntemlerinin geliřtirilmesi, daha ileri DDİ uygulamalarının hayata geirilmesini saęlayacaktır.

Sonuç olarak, Trk dilleri doęal dil iřleme alanında nemli bir potansiyele sahip olmakla birlikte, bu potansiyelin gerekleřtirilmesi, eřitli zorlukların ařılmasına baęlıdır. Bu makalede tartıřılan zorluklar ve zm nerileri, Trk dilleri zerine yapılan DDİ alıřmalarının daha iyi anlařılmasını saęlamakta ve bu alanda daha etkili zmler geliřtirilmesine katkıda bulunmayı amalamaktadır. Gelecekte yapılacak alıřmalar, Trk dillerinin DDİ alanındaki kullanımını daha da geniřletecek ve bu dillerin dilbilimsel arařtırmalar ve DDİ uygulamalarındaki yerini saęlamlařtıracaktır.

KAYNAKÇA

Abdurakhmonova, N. Z., Ismailov, A. S., & Mengliev, D. (2022, November). Developing NLP tool for linguistic analysis of Turkic languages. In 2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON) (pp. 1790-1793). IEEE.

Alkım, E., & Cebi, Y. (2019). Machine translation infrastructure for Turkic languages (MT-Turk). *The International Arab Journal of Information Technology*, 16(3), 380-388.

Bekeyeva, N., Bissengali, A., Mankeyeva, Z., & Nurdauletova, B. (2021). Phraseological Expressions in the Turkic Language: Comparative Analysis. *International Journal of Society, Culture & Language*, 9(2 (Themed Issue on Modern Realities of National Languages of CIS Countries)), 29-40.

Gatiatullin, A., Suleymanov, D., Prokopyev, N., & Khakimov, B. (2020, November). About Turkic Morpheme Portal. In *CMCL* (pp. 226-243).

ITU NLP. (2024). Language Resources and Tools for Turkic Languages. Erişim adresi: <https://nlp.itu.edu.tr/en/toolsandresources>. Erişim tarihi: 08.07.2024

Johanson, L. (2021). The structure of Turkic. In *The Turkic languages* (pp. 26-59). Routledge.

Kornfilt, J. (2018). Turkish and the Turkic languages. In *The world's major languages* (pp. 536-561). Routledge.

Kulamshae, K. (2019). Türk Lehçeleri Arasında Yapılacak Makine Çevirisine Kısa Bir Bakış: Başkurt Morfolojisinin İki

Seviyeli Tanımı Örneğinde. Sosyal Araştırmalar ve Davranış Bilimleri, 5(8), 469-483.

Madaan, A., Rijhwani, S., Anastasopoulos, A., Yang, Y., & Neubig, G. (2020). Practical comparable data collection for low-resource languages via images. arXiv preprint arXiv:2004.11954.

Magueresse, A., Carles, V., & Heetderks, E. (2020). Low-resource languages: A review of past work and future challenges. arXiv preprint arXiv:2006.07264.

Mengliev, D., Barakhnin, V., & Abdurakhmonova, N. (2021). Development of intellectual web system for morph analyzing of uzbek words. Applied Sciences, 11(19), 9117.

Mirzakhlov, J. (2021). Turkic interlingua: a case study of machine translation in low-resource languages (Master's thesis, University of South Florida).

Mirzakhlov, J., Babu, A., Ataman, D., Kariev, S., Tyers, F., Abduraufov, O., ... & Chellappan, S. (2021). A Large-Scale Study of Machine Translation in the Turkic Languages. arXiv preprint arXiv:2109.04593.

Mirzakhlov, J., Babu, A., Kunafin, A., Wahab, A., Moydinboyev, B., Ivanova, S., ... & Chellappan, S. (2021). Evaluating multiway multilingual NMT in the Turkic languages. arXiv preprint arXiv:2109.06262.

Mussakhoyeva, S., Dauletbek, K., Yeshpanov, R., & Varol, H. A. (2023). Multilingual speech recognition for Turkic languages. Information, 14(2), 74.

Oflazer, K. (2016). Türkçe ve Doğal Dil İşleme. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 5(2).

Ranathunga, S., Lee, E. S. A., Prifti Skenduli, M., Shekhar, R., Alam, M., & Kaur, R. (2023). Neural machine translation for low-resource languages: A survey. ACM Computing Surveys, 55(11), 1-37.

Safaya, A., Kurtuluş, E., Göktoğan, A., & Yuret, D. (2022). Mukayese: Turkish NLP strikes back. arXiv preprint arXiv:2203.01215.

Tantuğ, A. C., & Adalı, E. (2018). Machine translation between Turkic languages. Turkish Natural Language Processing, 237-254.

Tohma, K., & Kutlu, Y. (2020). Challenges Encountered in Turkish Natural Language Processing Studies. Natural and Engineering Sciences, 5(3), 204-211.

Tukeyev, U. (2023). A New Computational Model For Turkic Languages Morphology And Processing. Journal Of Problems in Computer Science And Information Technologies, 1(1).

Tukeyev, U., Karibayeva, A., Turganbayeva, A., & Amirova, D. (2021). Universal programs for stemming, segmentation, morphological analysis of Turkic words. In Computational Collective Intelligence: 13th International Conference, ICCCI 2021, Rhodes, Greece, September 29–October 1, 2021, Proceedings 13 (pp. 643-654). Springer International Publishing.

Zhanabergenova, D., & Tukeyev, U. (2021). Morphology model and segmentation for old Turkic language. In Computational

Collective Intelligence: 13th International Conference, ICCCI 2021, Rhodes, Greece, September 29–October 1, 2021, Proceedings 13 (pp. 629-642). Springer International Publishing.

BÖLÜM V

Farksal Gelişim (DE) Algoritması

Uğur TALAŞ

Giriş

Sezgisel optimizasyon algoritmaları genellikle doğadan esinlenen yöntemlerle bir fonksiyonun minimum yada maksimum değerini, belirlenen bir aralıkta sistematik bir şekilde bulmaya çalışır(Özsağlam & Çunkaş, 2008). Matematiksel olarak fonksiyon şeklinde ifade edebildiğimiz her problem sezgisel optimizasyon algoritmaları ile optimize edilebilir. Özellikle doğrusal olmayan ve yerel minimumlar içeren problemlerde klasik matematiksel yaklaşımlar çözüm uzayını taraması gerektiğinden çözüme ulaşmak yüksek maliyetli olmaktadır. Sezgisel yaklaşımlar ise kısa sürede tüm çözüm uzayını taramadan optimum çözüme yakın çözümler bulabilirler(Kokash, 2005).

Farksal Gelişim (Differential Evulation - DE) 1997 yılında Rainer storm ve Kenneth Price tarafından geliştirilmiştir(Storn &

Price, 1997). Temel olarak genetik algoritmalarındaki evrimsel prensiplere dayansa da kendine özgü bir yapısı vardır. Genetik algortmadan farklı olarak mutasyon, çaprazlama ve seçim sıralamasıyla mekanizmaları çalıştırır. Genetik algoritmalarda ise seçim, çaprazlama ve mutasyon sıralamasıyla mekanizmalar çalıştırılmaktadır. DE algoritması popülasyon tabanlı bir algortmadır. Popülasyon tabanlı algortmalar paralel işlem yapmaya uygun bir alt yapıları bulunmakta ve çeşitliliği yüksek olduğundan farklı tipteki problemlerde etkili çözümler bulabilirler(Beheshti & Shamsuddin, 2013). Bu sebeple bir çok farklı çalışma ve günlük hayat probleminde DE algoritması kullanılmıştır.

Y. YAŞAR ve B. DURMUŞ yaptıkları “Farksal Gelişim Algoritması İle Karma Yem Maliyet Optimizasyonu” isimli çalışmalarında Tavşan büyüme yemi için düşük maliyetli uygun yem karışımlarının üretilmesi süreçlerini optimize etmişlerdir. Optimize işlemi için Parçacık Sürü Optimizasyonu (PSO), Gerçek Kodlu Genetik Algoritma (RGA) ve Lineer Programlama (LP) ile Farksal Gelişim algortmalarının sonuçlarını kıyaslamışlardır. Yapılan çalışmaya göre en başarılı sonucu DE algoritması vermiştir. (Yaşar & Durmuş, 2014)

İ.DEVELİ ve E.N.YAZLIK yaptıkları bir çalışmada çoklu antenlere sahip olan Çok Girişli Çok Çıkışlı sistemler üzerinde ergodik kapasiteyi optimize eden optimum anten konfigürasyonunu için optimizasyon işlemi gerçekleştirmişlerdir. Optimizasyon sürecinde Farksal Gelişim optimizasyon algortmasından yararlanmışlardır. Çalışmada denedikleri klasik ayrıntılı araştırma metotları ya da tekrarlı denemeli metotlara kıyasla, FG algortması

çözüm süresi ve problemin çözülebilirliği açısından çok daha etkili olduğunu söylemişlerdir.(Yazlık & Develi, 2009)

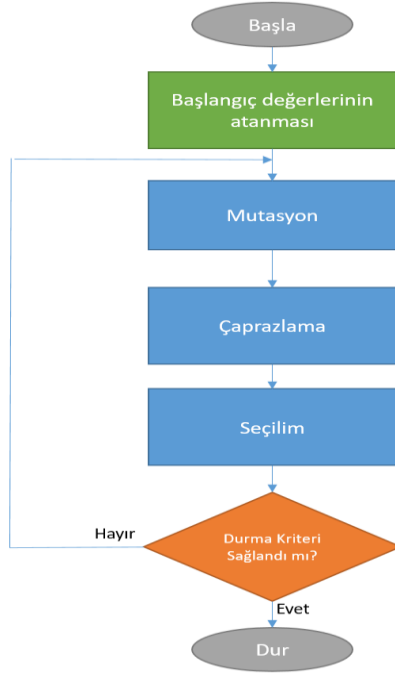
Yapılan bir çalışmada Farksal Gelişim Algoritması kullanılarak kısmi iletim dizisi yöntemiyle dikgen frekans bölmeli çoğullama sistemlerinde tepe gücü/ortalama güç oranının düşürülmesi üzerine çalışılmıştır. Geleneksel kısmi iletim dizisi tekniği karmaşık hesaplama gerektirdiğinden sezgisel bir yaklaşım kullanılmıştır. Önerilen yöntem geleneksel yöntemle, parçacık sürü optimizasyonu ve genetik algoritmalarla birlikte kıyaslanmıştır. Karşılaştırma sonucunda en iyi sonucu farksal gelişim optimizasyon algoritmasının verdiği ifade edilmiştir. (Bozkurt & Taşpınar, 2014)

DE algoritması farklı problemlerin çözümü için literatürde yaygın kullanılan algoritmalarından biridir(Plagianakos Ark., 2008). Özellikle kolay uygulanabilirliği ve az parametre içermesi, basit matematiksel alt yapısı sayesinde çok sayıda çalışmada kullanılmıştır. Yazılımsal olarak kodlaması da oldukça basit olan bu algoritma bir sonraki bölümde çalışma adımları ve matematiksel modeli incelenip, devam eden bölümde $f(x) = x^2 + y^2$ örnek problemi python dili ile kodlaması yapıp her bir iterasyondaki elde edilen ölçümler verilmiş hem de grafik üzerinde global optimuma yakınsaması incelenmiştir. (Desale Ark., 2015)Şekil, grafik, diagram vb. gibi çizim ya da veri güncelleme isteyen unsurlar JPEG olarak eklenmelidir.

Farksal Gelişim Optimizasyon Algoritması (DE)

DE algoritması Rainer storm ve Kenneth Price tarafından 1997 yılında ortaya atılmıştır.(Storn & Price, 1997) 27 yıl önce geliştirilen bu algoritma bir çok mühendislik probleminde ve algoritmaları

kıyaslamak için kullanılan benchmark problemlerinde kullanılmıştır(Sepesy Maučec & Brest, 2019). Genel olarak genetik algoritmalarındaki evrimsel prensiplere dayan bu algoritma mutasyon, çaprazlama ve seçim mekanizmalarını kendine özgü bir yöntemle uygular(Opara & Arabas, 2019). Bu mekanizmalarda kullanılan parametre ve matematiksel yöntemin sadeliği kullanılabilirliğini arttırmaktadır. Ayrıca kararlı bir algoritma olması çok tercih edilmesini sağlamaktadır. Algoritmanın akış diyagramı aşağıdaki Şekil 1’de verilmiştir.



Şekil 2: DE akış diyagramı

Şekil 1’de sunulan diyagramında DE’nin çalışma mekanizmaları sırasıyla akış olarak verilmiştir. İlk olarak başlangıç

parametreleri belirlenip başlangıç popülasyonu rastgele oluşturulur. Sonrasında popülasyonu oluşturan her bir birey için mutasyon çaprazlama ve seçim işlemleri gerçekleştirilir. Mutasyon işleminde fark vektörü hesaplanır, çaprazlamada başlangıçta belirlenen çaprazlama oranı (Crossover Rate - CR) olasılığıyla çaprazlama yapılır ve son olarak seçimde başlangıçtaki birey ile çaprazlama sonucunda oluşan aday bireylerin fitness değerleri karşılaştırılır. Bu işlem durma kriterine kadar iteratif bir şekilde tekrarlanır ve durma kriterine ulaşıldığında algoritma durdurulur. Bu çalışma akışı aşağıda adımlar halinde detaylı olarak verilmiştir.

1. Adım: Başlangıç parametrelerinin belirlenmesi

DE algoritması uygulanırken popülasyon boyutu (NP), Çaprazlama oranı (Crossover Rate - Cr), ve Faktör (F) değerleri belirlenir. Cr ve F parametreleri 0 ile 1 aralığında belirlenir. Bu değerlerin ne olması gerektiği probleme özgü farklı değerler denenerek belirlenebilir.

2. Adım: Popülasyonun oluşturulması

Başlangıçta belirlenen NP değerine göre rastgele bireyler üretilir. Rastgele bireyler üretilirken çözüm kümesi için belirlenen kısıtlara uygun şekilde belirlenir.

3. Adım: Mutasyon

Popülasyondaki her birey için, popülasyondan rastgele seçilen üç bireyi kullanarak bir fark vektörü hesaplanır. Fark vektörü formülü aşağıdaki Eşitlik 1'de verilmiştir.

$$V = X_{r1} + F * (X_{r2} - X_{r3}) \quad (1)$$

V : Fark Vektörü,

F : Faktör,

X_{r1}, X_{r2}, X_{r3} : Popülasyondan rastgele seçilen üç birey

Eşitlik 1’de sunulan matematiksel formül popülasyondaki her bir birey için uygulanırken eğer popülasyon birden fazla boyutlu ise her bir boyut içinde ayrı ayrı hesaplanmaktadır.

4. Adım: Çaprazlama

Çaprazlama işlemi aday çözüm ve mevcut çözümün belirli bir rastgelelik oranıyla harmanlaması işlemidir. Çaprazlama işleminin hangi oranla yapılacağı birinci adımda belirlenen Cr ’ye bağlıdır. Cr parametresi $[0,1]$ aralığında değer alırken 1’ e yakın seçilmesi çaprazlanma olasılığını arttırırken 0’a yakın seçilmesi ise çaprazlama olasılığını azaltmaktadır. Çaprazlama işlemi aşağıdaki Eşitlik 2’de kurala göre yapılmaktadır.

$$U_{i,j} = \begin{cases} V_{i,j} & \text{Eğer } rand(0,1) < Cr \\ X_{i,j} & \end{cases} \quad (2)$$

$V_{i,j}$: aday çözümü temsil eder

$X_{i,j}$: mevcut çözümü temsil eder

Cr : çaprazlama oranını temsil eder

Eşitlik 2’de verilen kurala göre algoritma çalışma anında rastgele bir sayı üretir ve bu sayının Cr ’den küçük olma yada büyük olma durumuna göre aday çözüm ile mevcut çözümü harmanlar. Bu işlem çok boyutlu problemlerde her bir boyut için ayrı ayrı yapılır.

5.Adım: Seçilim

Seçilim adımında aday birey ile mevcut bireyden hangisinin popülasyona devam edeceğine karar verilir. Her iki bireyinde fitness değerleri karşılaştırılır problemin çözümü için daha uygun olan çözüme sahip birey popülasyonda dahil edilir ve seçim işlemi tamamlanır.

6.Adım: Durdurma kontrolü

Durdurma 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 3. Adıma tekrar dönülerek yapılan işlemler tekrar edilir.

DE algoritmasının yukarıda 6 adımda işlem detayları verilmiştir. DE'nin çalışma adımları algoritması aşağıdaki Algoritma 1 'de sunulmuştur.

Algoritma 1: Farksal Gelişim Optimizasyon Algoritması

Başla:

1. Popülasyonu başlat:
2. Her birey için fitness değerlerini hesapla.
3. En iyi bireyi seç (fitness değeri en düşük olan)
4. İterasyon başlat: (Durma kriterine kadar devam ettir.)

4.1. Her birey için:

a. Mutasyon:

- Rastgele üç farklı birey seç : X_{r1}, X_{r2}, X_{r3}
- Yeni birey (mutant) oluştur: $V = X_{r1} + F * (X_{r2} - X_{r3})$

b. Çaprazlama:

- Mevcut birey ($X_{i,j}$) ile mutant ($V_{i,j}$) arasında çaprazlama yap.

$$U_{i,j} = \begin{cases} V_{i,j} & \text{Eğer } rand(0,1) < Cr \\ X_{i,j} & \end{cases}$$

c. Seçim:

- Çaprazlama sonucunda oluşan bireyin fitness değeri $f(U_{i,j})$ ile mevcut bireyin fitness $f(X_{i,j})$ değeri karşılaştırılır. Uygun olan popülasyona dahil edilir.

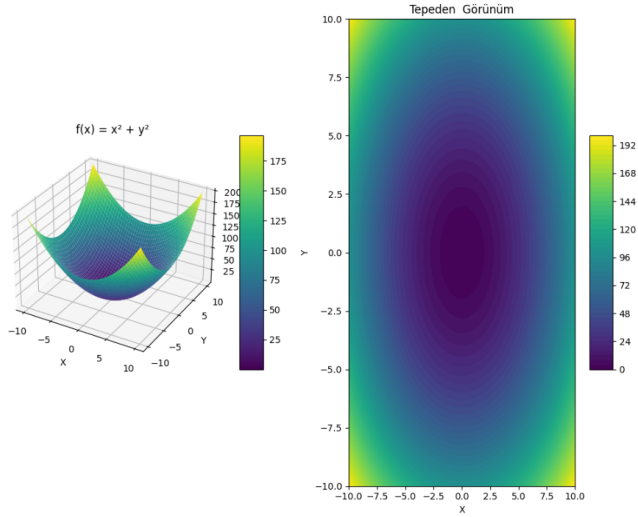
6. En iyi çözümü güncelle
7. Algoritma tamamlandığında, en iyi çözümü ve ilgili fitness değerini döndür.

Bitiş

DE basit matematiksel işlemleri sayesinde ve sade algoritması ile kolay uygulanabilir bir algoritmadır. Çalışmanın devam eden bölümlerinde $f(x) = x^2 + y^2$ örneği üzerinde python kodu ve uygulamada elde edilen sonuçlar grafik üzerinde sunulmuştur.

Örnek 1 : DE $f(x) = x^2 + y^2$ Örneği

$f(x) = x^2 + y^2$ fonksiyonunun minimum değerini bulmak için python dili ile DE algoritması yazılmış ve her iterasyon sonundaki sonuçlar ve popülasyonun grafik üzerindeki son durumu incelenmiştir. Problemi oluşturan $f(x) = x^2 + y^2$ fonksiyonunun grafiği aşağıdaki Şekil 2’de verilmiştir.



Şekil 3: $f(x) = x^2 + y^2$ grafiği

Algoritmanın başlangıç parametreleri;

$$N_p = 5 \quad F = 0,8 \quad Cr = 0,7$$

Olarak belirlenmiştir.

Python kodu:

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

# Parametreler
pop_size = 5          # Popülasyon boyutu
dimensions = 2
mutation_factor = 0.8
crossover_rate = 0.7
iterations = 10      # İterasyon sayısı

# Hedef fonksiyon  $f(x, y) = x^2 + y^2$ 
def objective_function(x):
    return x[0] ** 2 + x[1] ** 2

# Başlangıç popülasyonu oluştur (2D)
population = np.random.uniform(-5.12, 5.12, (pop_size, dimensions))

# Her iterasyonda işlemleri ekrana yazdır ve grafiği çizdir
for i in range(iterations):
    print(f"\nIteration {i + 1}")
    new_population = np.copy(population)
    print(f"Popülasyon: {new_population}")
    print("\n")

    # Bireyleri işleyin
    for j in range(pop_size):
        # Üç rastgele farklı birey seç
        indices = np.random.choice(pop_size, 3, replace=False)
        while j in indices:
            indices = np.random.choice(pop_size, 3, replace=False)
        print("İşlem yapılan birey ", population[j])
        a, b, c = population[indices]
        mutant = a + mutation_factor * (b - c)
        print("Rastgele seçilen 3 birey : xr1", a, " xr2", b, "
xr3", c )
        # Mutasyon ve çaprazlama
        print("Fark Vektörü : ", mutant)

        trial = np.copy(population[j])

        for d in range(dimensions):
            rn = np.random.rand()
            print("Rastgele Sayı : ", rn, " Cr : ", crossover_rate)
            if rn < crossover_rate:
                trial[d] = mutant[d]

    ui = trial
    print("ui : ", ui)

# Değerlendirme
```

```

        if objective_function(trial) <
objective_function(population[j]):
            new_population[j] = trial

    # Hesaplamaları yazdır
    print(f"Birey {j + 1}: Xi = {population[j]}, ui = {trial},
"
        f" xi Fitness=
{objective_function(population[j]):.4f},"
        f" ui Fitness = {objective_function(trial):.4f}")
    print("\n\n")

# Popülasyonu güncelle
population = new_population

# Grafik çizimi için hazırlık
x_values = np.linspace(-5.12, 5.12, 100)
y_values = np.linspace(-5.12, 5.12, 100)
X, Y = np.meshgrid(x_values, y_values)
Z = X ** 2 + Y ** 2 # f(x, y) = x^2 + y^2

# Grafik çizimi
fig = plt.figure(figsize=(18, 6))

# 3D Surface plot
ax1 = fig.add_subplot(131, projection='3d')
ax1.plot_surface(X, Y, Z, cmap='viridis', edgcolor='none',
alpha=0.7)
ax1.scatter(population[:, 0], population[:, 1],
[objective_function(ind) for ind in population], color='red')
ax1.set_title(f'3D (İterasyon : {i + 1})')
ax1.set_xlabel('X')
ax1.set_ylabel('Y')
ax1.set_zlabel('Fitness')

# Contour plot
ax2 = fig.add_subplot(132)
contour = ax2.contourf(X, Y, Z, levels=50, cmap='viridis')
fig.colorbar(contour, ax=ax2)
ax2.scatter(population[:, 0], population[:, 1], color='red',
label='Birey')
ax2.set_title(f'Tepeden Görünüm (İterasyon {i + 1})')
ax2.set_xlabel('X')
ax2.set_ylabel('Y')
ax2.legend()

# 2D Scatter plot of population on the function f(x, y) = x^2 +
y^2
ax3 = fig.add_subplot(133)
z_values = [objective_function([x, y]) for x, y in population]
ax3.plot(x_values, [objective_function([x, 0]) for x in
x_values], label='Objective Function (1D Slice)', color='orange')
ax3.scatter(population[:, 0], z_values, color='blue',
label='Birey')

```

```

ax3.set_title(f'Kesiti alınmış grafik (İterasyon {i + 1})')
ax3.set_xlabel('X')
ax3.set_ylabel('Fitness')
ax3.legend()
ax3.grid(True)

plt.show()

# Sonuç
best_individual = min(population, key=objective_function)
print("\n En iyi çözüm:", best_individual)
print("En iyi fitness:", objective_function(best_individual))

```

DE $f(x) = x^2 + y^2$ örneği sonuçlarının iteratif olarak incelenmesi:

Algoritmanın $f(x) = x^2 + y^2$ problemi için sonuçları bu bölümde verilmiştir. Sonuçlar verilirken 1. iterasyondaki tüm hesaplama çıktıları verilmiştir ve tekrar eden iterasyonlarda sadece popülasyondaki bireyler ve bireylerin grafikler üzerindeki pozisyonları verilmiştir. Grafikler üzerinden incelendiğinde popülasyonun optimizasyon süreci adım adım izlenilebilmektedir.

1. İterasyon

Popülasyon:

Tablo 3: Birinci iterasyondaki popülasyon

	X	Y
1	-2.471862	-4.30521851
2	1.6693909	-3.34790286
3	0.98881984	-3.97888436
4	4.16981518	4.46110603
5	-3.58451359	-0.15652833

1. Birey: $X_1 = [-2.471862 \quad -4.30521851]$

Mutasyon işlemi :

Rastgele seçilen bireyler : $X_{r1} = [1.6693909 \quad -3.34790286]$
 $X_{r2} = [-3.58451359 \quad -0.15652833]$
 $X_{r3} = [4.16981518 \quad 4.46110603]$

Hesaplanan Fark vektörü $V = [-4.53407212 \ -7.04201035]$

Çaprazlama :

X boyutu için rastgele sayı: 0.95 Cr : 0.7

X boyutu için rastgele sayı: 0.97 Cr : 0.7

Aday Birey $U_{i,j} = [-2.471862 \ -4.30521851]$

Seçilim :

$f(X_{i,j}) = 24.6450$ * Seçilmiştir.

$f(U_{i,j}) = 24.6450$

2.Birey: $X_2 = [1.6693909 \ -3.34790286]$

Mutasyon işlemi :

Rastgele seçilen bireyler : $X_{r1} = [4.16981518 \ 4.46110603]$

$X_{r2} = [-2.471862 \ -4.30521851]$

$X_{r3} = [0.98881984 \ -3.97888436]$

Hesaplanan Fark vektörü $V = [1.40126971 \ 4.20003871]$

Çaprazlama :

X boyutu için rastgele sayı: 0.44 Cr : 0.7

X boyutu için rastgele sayı: 0.18 Cr : 0.7

Aday Birey $U_{i,j} = [1.40126971 \ 4.20003871]$

Seçilim :

$f(X_{i,j}) = 13.9953$ * Seçilmiştir.

$f(U_{i,j}) = 19.6039$

3.Birey: $X_3 = [0.98881984 \ -3.97888436]$

Mutasyon işlemi :

Rastgele seçilen bireyler : $X_{r1} = [1.6693909 \ -3.34790286]$

$X_{r2} = [-3.58451359 \ -0.15652833]$

$X_{r3} = [-2.471862 \ -4.30521851]$

Hesaplanan Fark vektörü $V = [0.77926962 \ -0.02895072]$

Çaprazlama :

X boyutu için rastgele sayı: 0.19 Cr : 0.7

X boyutu için rastgele sayı: 0.77 Cr : 0.7

Aday Birey $U_{i,j} = [0.77926962 \ -3.97888436]$

Seçilim :

$$f(X_{i,j}) = 16.8093$$

$$f(U_{i,j}) = 16.4388 * \text{Seçilmiştir.}$$

$$4.\text{Birey: } X_4 = [4.16981518 \ 4.46110603]$$

Mutasyon işlemi :

$$\text{Rastgele seçilen bireyler : } X_{r1} = [-2.471862 \ -4.30521851]$$

$$X_{r2} = [-2.471862 \ -4.30521851]$$

$$X_{r3} = [1.6693909 \ -3.34790286]$$

$$\text{Hesaplanan Fark vektörü } V = [-6.89751591 \ -0.92238085]$$

Çaprazlama :

$$X \text{ boyutu için rastgele sayı: } 0.53 \quad Cr : 0.7$$

$$X \text{ boyutu için rastgele sayı: } 0.26 \quad Cr : 0.7$$

$$\text{Aday Birey } U_{i,j} = [-6.89751591 \ -0.92238085]$$

Seçilim :

$$f(X_{i,j}) = 37.2888 * \text{Seçilmiştir.}$$

$$f(U_{i,j}) = 48.4265$$

$$5.\text{Birey: } X_5 = [-3.58451359 \ -0.15652833]$$

Mutasyon işlemi :

$$\text{Rastgele seçilen bireyler : } X_{r1} = [1.6693909 \ -3.34790286]$$

$$X_{r2} = [-3.58451359 \ -0.15652833]$$

$$X_{r3} = [-2.471862 \ -4.30521851]$$

$$\text{Hesaplanan Fark vektörü } V = [0.77926962 \ -0.02895072]$$

Çaprazlama :

$$X \text{ boyutu için rastgele sayı: } 0.19 \quad Cr : 0.7$$

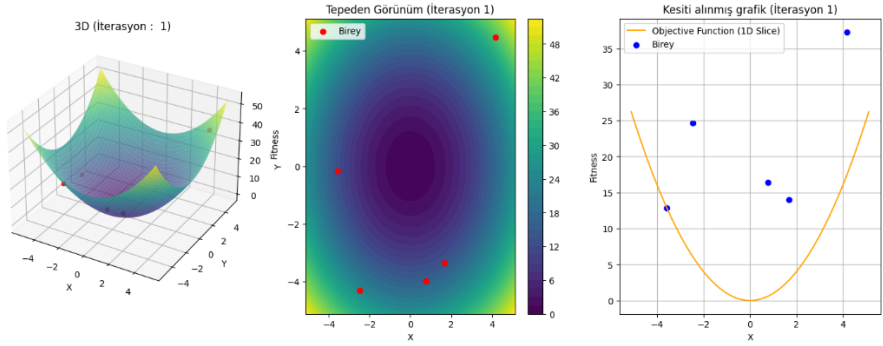
$$X \text{ boyutu için rastgele sayı: } 0.77 \quad Cr : 0.7$$

$$\text{Aday Birey } U_{i,j} = [0.77926962 \ -3.97888436]$$

Seçilim :

$$f(X_{i,j}) = 16.8093$$

$$f(U_{i,j}) = 16.4388 * \text{Seçilmiştir.}$$

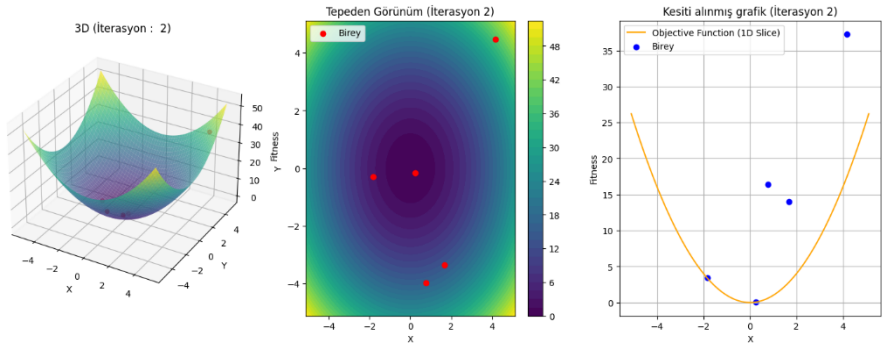


Şekil 4: Birinci iterasyon sonunda popülasyonun grafik üzerinde gösterimi

2. İterasyon

Tablo 4: İkinci iterasyondaki popülasyon

	X	Y
1	-2.471862	-4.30521851
2	1.6693909	-3.34790286
3	0.77926962	-3.97888436
4	4.16981518	4.46110603
5	-3.58451359	-0.15652833

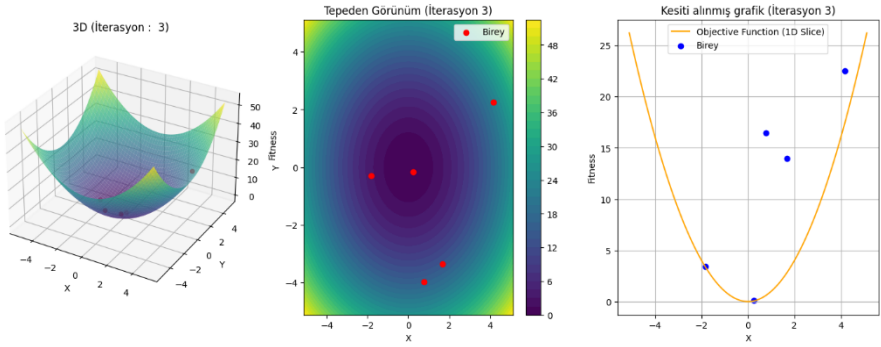


Şekil 5: İkinci terasyon sonunda popülasyonun grafik üzerinde gösterimi

3. İterasyon

Tablo 5: Üçüncü iterasyondaki popülasyon

	X	Y
1	-1.82163567	-0.29001804
2	1.6693909	-3.34790286
3	0.77926962	-3.97888436
4	4.16981518	4.46110603
5	0.24057445	-0.15652833

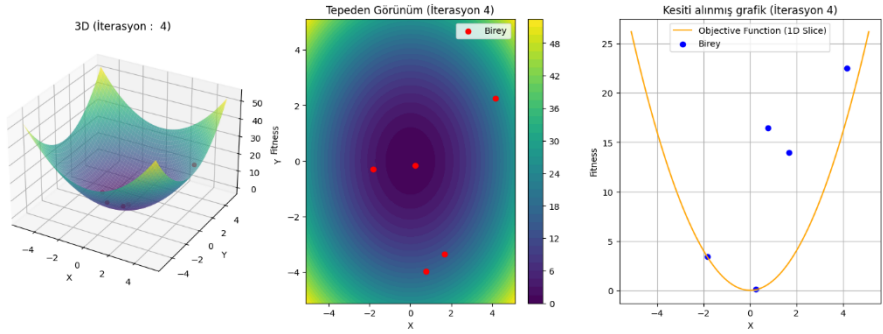


Şekil 6: Üçüncü iterasyon sonunda popülasyonun grafik üzerinde gösterimi

4. İterasyon

Tablo 6: Dördüncü iterasyondaki popülasyon

	X	Y
1	-1.82163567	-0.29001804
2	1.6693909	-3.34790286
3	0.77926962	-3.97888436
4	4.16981518	2.26308158
5	0.24057445	-0.15652833

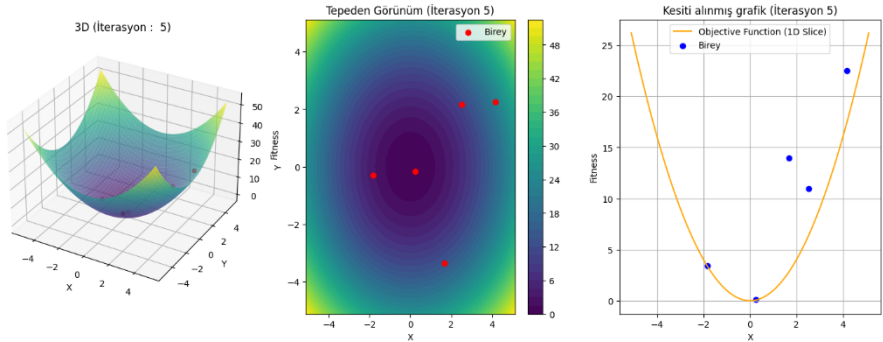


Şekil 7: Dördüncü iterasyon sonunda popülasyonun grafik üzerinde gösterimi

5. İterasyon

Tablo 7: Beşinci iterasyondaki popülasyon

	X	Y
1	-1.82163567	-0.29001804
2	1.6693909	-3.34790286
3	0.77926962	-3.97888436
4	4.16981518	2.26308158
5	0.24057445	-0.15652833

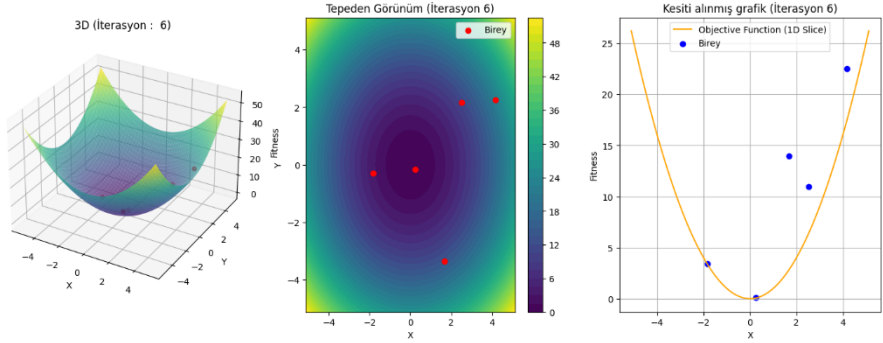


Şekil 8: Beşinci iterasyon sonunda popülasyonun grafik üzerinde gösterimi

6. İterasyon

Tablo 8: Altıncı iterasyondaki popülasyon

	X	Y
1	-1.82163567	-0.29001804
2	1.6693909	-3.34790286
3	2.52004709	2.15628981
4	4.16981518	2.26308158
5	0.24057445	-0.15652833

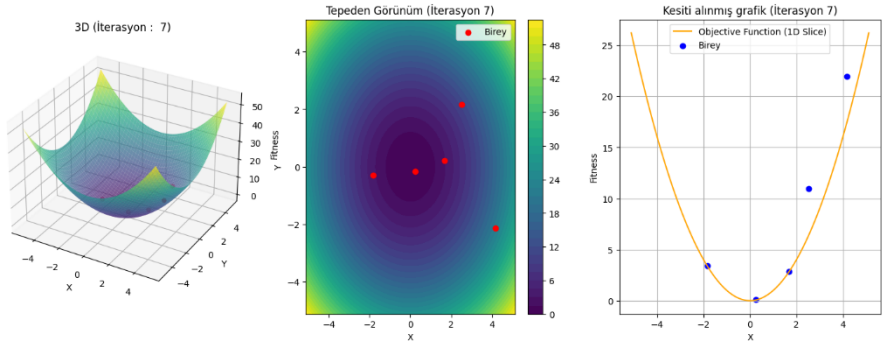


Şekil 9: Altıncı iterasyon sonunda popülasyonun grafik üzerindeki gösterimi

7. İterasyon

Tablo 9: Yedinci iterasyondaki popülasyon

	X	Y
1	-1.82163567	-0.29001804
2	1.6693909	-3.34790286
3	2.52004709	2.15628981
4	4.16981518	2.26308158
5	0.24057445	-0.15652833

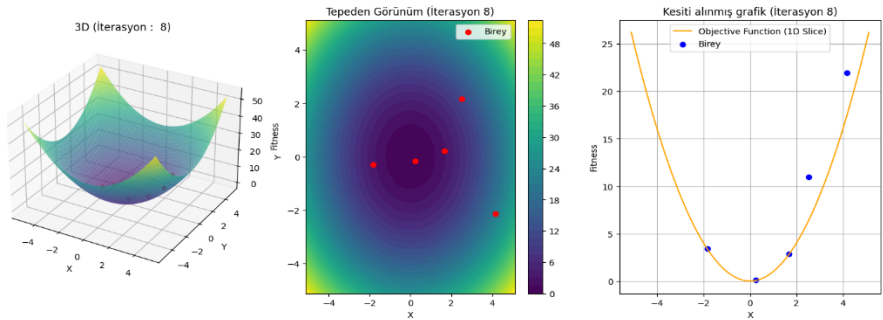


Şekil 10: Yedinci iterasyon sonunda popülasyonun grafik üzerinde gösterimi

8. İterasyon

Tablo 10: Sekizinci iterasyondaki popülasyon

	X	Y
1	-1.82163567	-0.29001804
2	1.6693909	0.22060188
3	2.52004709	2.15628981
4	4.16981518	-2.14027256
5	0.24057445	-0.15652833

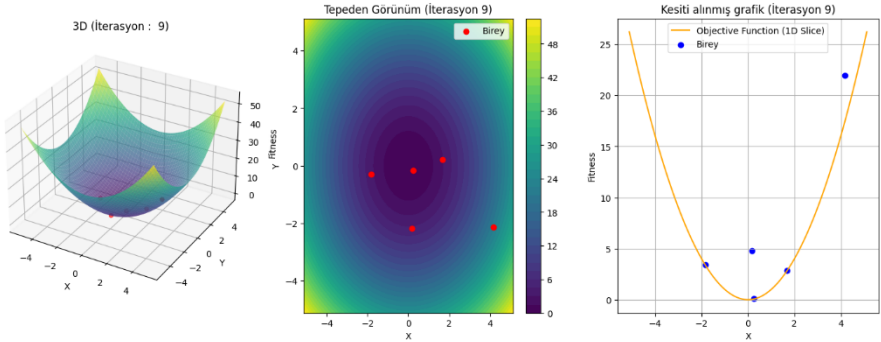


Şekil 11: Sekizinci iterasyon sonunda popülasyonun grafik üzerinde gösterimi

9. İterasyon

Tablo 11: Dokuzuncu iterasyondaki popülasyon

	X	Y
1	-1.82163567	-0.29001804
2	1.6693909	0.22060188
3	2.52004709	2.15628981
4	4.16981518	-2.14027256
5	0.24057445	-0.15652833

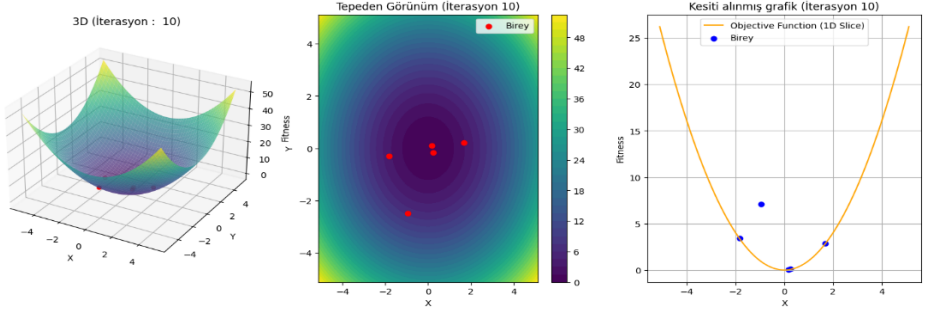


Şekil 12: Dokuzuncu iterasyon sonunda popülasyonun grafik üzerinde gösterimi

10. İterasyon

Tablo 12: Onuncu iterasyondaki popülasyon

	X	Y
1	-1.82163567	-0.29001804
2	1.6693909	0.22060188
3	0.17870375	-2.17871759
4	4.16981518	-2.14027256
5	0.24057445	-0.15652833



Şekil 13: Onuncu iterasyon sonunda popülasyonun grafik üzerinde gösterimi

En iyi çözüm: [0.24057445 -0.15652833]

En iyi fitness: 0.08237718408

Örnek 10. İterasyonda durma kriterine ulaştığından sonlanmıştır. 10. İterasyon sonunda en uygun çözüm [0.24057445 - 0.15652833] olarak bulunmuştur.

Bu çalışmada DE algoritmasının genel yapısı incelenmiştir. Çalışma adımları çıkartılarak matematiksel altyapısı, akış diyagramı verilmiş ve literatürdeki çalışmalardan bahsedilmiştir. Son olarak $f(x) = x^2 + y^2$ örneği üzerinden DE algoritmasının optimizasyon süreci grafikler üzerinden sunulmuştur. Birinci ve ikinci iterasyonda hesaplama adımları verilen algoritma 3 iterasyondan itibaren popülasyonun güncel durumu ve grafik üzerinde popülasyondaki bireylerin yerleri işaretlenmiştir. Bu sayede algoritmanın iteratif bir şekilde nasıl sonuca gittiği gözlemlenmiştir.

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.

BOZKURT Yüksel, & TAŞPINAR Necmi. (2014, Ağustos 28). Farksal Gelişim Algoritması kullanılarak Kısmi İletim Dizisi Yöntemiyle Dikgen Frekans Bölmeli Çoğullama Sistemlerinde Tepe Gücü/Ortalama Güç Oranının Düşürülmesi. *URSİ Türkiye 2014 VII. Bilimsel Kongresi*.

Desale, S., Rasool, A., Andhale, S., & Rane, P. (2015). Heuristic and meta-heuristic algorithms and their relevance to the real world: a survey. *Int. J. Comput. Eng. Res. Trends*, 351(5), 2349-7084.

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

Opara, K. R., & Arabas, J. (2019). Differential Evolution: A survey of theoretical analyses. *Swarm and evolutionary computation*, 44, 546-558.

Özsağlam, M. Y., & Çunkaş, M. (2008). Optimizasyon Problemlerinin Çözümü için Parçaçık Sürü Optimizasyonu Algoritması. *Journal of Polytechnic*, 11(4), 299-305. <https://dergipark.org.tr/en/pub/politeknik/issue/33033/367241>

Plagianakos, V. P., Tasoulis, D. K., & Vrahatis, M. N. (2008). A review of major application areas of differential evolution. *Advances in differential evolution*, 197-238.

Sepesy Mauec, M., & Brest, J. (2019). A review of the recent use of Differential Evolution for Large-Scale Global Optimization: An analysis of selected algorithms on the CEC 2013 LSGO benchmark suite. *Swarm and Evolutionary Computation*, 50, 100428. <https://doi.org/10.1016/J.SWEVO.2018.08.005>

Storn, R., & Price, K. (1997). Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4), 341-359. <https://doi.org/10.1023/A:1008202821328/METRICS>

Yařar, Y., & Durmuř, B. (2014). Farksal Geliřim Algoritması İle Karma Yem Maliyet Optimizasyonu. *Journal of Science and Technology of Dumlupınar University* (Sayı 033, ss. 33-44). Kütahya Dumlupınar University. <https://dergipark.org.tr/en/pub/dpufbed/issue/35917/403301>

Yazlık, E. N., & Develi, İ. (2009). *Farksal geliřim algoritmasının ok giriřli ok ıkıřlı sistemlerde optimum anten konfigürasyonuna uygulanması*. <http://acikerisim.nevsehir.edu.tr/xmlui/handle/20.500.11787/6251>

BÖLÜM VI

Derin Öğrenme Teknikleri ve Uygulamaları

Alper Talha KARADENİZ¹

1. Giriş

Günümüzde derin öğrenme, yapay zeka ve makine öğrenimi devrim niteliğinde bir teknoloji haline gelmiştir. Çok katmanlı sinir ağlarının sağladığı yüksek doğruluk ve esneklik, büyük veri kümelerinin işlenmesi ve karmaşık problemlerin çözülmesi gibi konularda bu yöntemleri daha da popüler hale getirmiştir (LeCun ve ark., 2015). Derin öğrenmenin en önemli özelliği, ham verilerdeki karmaşık desenleri ve ilişkileri otomatik olarak öğrenme yeteneğidir. Bu yetenek sayesinde derin öğrenme yöntemleri, görüntü işleme, doğal dil işleme, sağlık, finans, otomotiv ve tarım gibi birçok sektörde yaygın olarak kullanılmaktadır. Derin öğrenmenin temeli, çok katmanlı yapay sinir ağlarının öğrenme modeline ve soyutlama fikrine dayanmaktadır. Teknolojinin gelişmesi, cihazların donanım kapasitesi ve işlem gücünün artması ile birlikte, derin öğrenme yöntemleri çok çeşitli uygulamalar ve farklı sektörlerde yer

¹ Dr. Öğr. Üyesi Samsun Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Yazılım Mühendisliği Bölümü

almaktadır (Schmidhuber, 2015). Özellikle grafik işleme birimleri (GPU) ve dağıtık hesaplama teknikleri gibi modern donanım çözümleri, derin öğrenme algoritmalarının daha verimli bir şekilde eğitilmesine ve daha yüksek doğruluk oranlarıyla sonuçlar elde edilmesine olanak tanımaktadır (Buckner, 2019).

Derin öğrenme teknolojilerinin gelişmesi farklı sinir ağı mimarilerinin ortaya çıkmasına yol açmaktadır. Evrimsel Sinir Ağları (CNN), Tekrarlayan Sinir Ağları (RNN), Generative Adversarial Networks (GAN) ve Vision Transformer (ViT) gibi mimariler, farklı veri türleri ve farklı uygulamalar için geliştirilen mimarilerdir. Bu mimarilerin her biri, belirli görevlerde yüksek performans sağlayarak, farklı alanlardaki zorlukların üstesinden gelinmesine yardımcı olmak için geliştirilmiştir. Örneğin, CNN'ler görüntü işleme ve sınıflandırma uygulamalarında popüler hale gelmiştir. RNN'ler sıralı verilerin işlenmesi, GAN'ler veri oluşturma ve artırma işlemleri, ViT büyük veri kümeleri üzerinde paralel işlem yapılması konularında öne çıkmaktadır (Goodfellow, 2016). Bu çeşitlilik, derin öğrenme teknolojisinin farklı alanlarda ve sektörlerde başarılı bir şekilde kullanılmasını sağlamaktadır.

1.1. Derin öğrenmenin tarihsel gelişimi

Derin öğrenmenin temelleri, yapay sinir ağlarının ilk ortaya çıktığı 1980'lere dayanmaktadır. Sinir ağlarının katman sayısının artmasıyla birlikte derin öğrenme kavramı ortaya çıkmış ve özellikle büyük veri setlerinin analiz edilmesine olanak tanımıştır (Schmidhuber, 2015). Gelişen teknolojiyle birlikte, veri işleme kapasitesi ve hesaplama gücündeki artış, derin öğrenme teknolojilerinin kullanımını daha geniş alanlara yaymış ve daha yüksek doğruluk oranlarıyla sonuçlanan modellerin geliştirilmesine katkıda bulunmuştur (Buckner, 2019).

1.2. Derin öğrenmenin tanımı

Derin öğrenme, çok katmanlı yapay sinir ağları aracılığıyla ham verilerden anlamlı özellikler çıkarmaya odaklanan bir makine öğrenmesi yöntemidir. Her katmanda gerçekleştirilen öğrenme

süreci, verilerin daha soyut temsillerini oluşturmakta ve karmaşık yapıları anlamaya olanak tanımaktadır. Bu özellik, derin öğrenme algoritmalarının görüntü, metin, ses ve diğer yüksek boyutlu veri türlerinde büyük başarı elde etmesini sağlamaktadır (Goodfellow, 2016).

2. Literatür incelemesi

Literatür, farklı derin öğrenme mimarileri, optimizasyon teknikleri ve mimarilerin uygulama alanlarındaki gelişimini kapsamlı bir şekilde ele almaktadır. Bu bölümde, mevcut çalışmaların genel bir değerlendirilmesi yapılmaktadır.

Evrışimsel Sinir Ağları (CNN), Tekrarlayan Sinir Ağları (RNN), Generative Adversarial Networks (GAN) ve Vision Transformer (ViT) gibi popüler mimariler, farklı uygulama alanlarında sıklıkla kullanılmaktadır. Bu mimarilerin farklı alanlardaki çeşitli uygulama örnekleri, derin öğrenmenin esnekliğini ve başarısını gözler önüne sermektedir. Mevcut literatüre dayanarak, bu mimarilerin birçok farklı uygulama alanında nasıl kullanıldığını gösteren uygulama örnekleri aşağıda sunulmaktadır.

CNN'ler, özellikle görüntü işleme ve sınıflandırma konularında yaygın olarak kullanılmaktadır. Karadeniz ve arkadaşları tarafından yapılan çalışmada, artık blok tabanlı CNN modeli, ceviz yapraklarından elde edilen görüntülerin sınıflandırılmasında kullanılmış ve yüksek doğruluk sağlamıştır. Bu çalışma, CNN'in tarım alanında etkili bir şekilde uygulanabileceğini göstermektedir (Karadeniz, Başaran& Çelik., 2023)

CNN'in sağlık alanındaki uygulamaları da dikkat çekmektedir. Başaran tarafından yürütülen bir çalışmada, beyin tümörü teşhisi için CNN özellikleri ve optimizasyon algoritmaları kullanılmıştır. Çalışmada, metin tabanlı özellik çıkarma algoritmalarının CNN ile birleştirilmesi sonucunda, teşhis doğruluğunun önemli ölçüde artırıldığı görülmüştür (Başaran, 2022).CNN'in sağlık alanında tıbbi görüntülerin sınıflandırılmasında yaygın olarak kullanıldığı bir başka örnek de, cilt kanseri

sınıflandırması üzerine yapılan çalışmalardır. Bu çalışmalarda, görüntü sınıflandırma görevinde CNN tabanlı modellerin başarılı sonuçlar verdiği gösterilmiştir (Esteva ve ark., 2019).

CNN'in endüstriyel uygulamalara entegrasyonu da oldukça yaygındır. Kalite kontrol süreçlerinde, üretim hatlarındaki görüntülerin analiz edilmesi ve nesnelerin tanınması için CNN tabanlı modeller kullanılmaktadır. Bu sayede, üretim süreçlerinin otomatikleştirilmesi ve hatalı ürünlerin tespit edilmesi mümkün olmaktadır (Zeebaree ve ark., 2020).

RNN ve LSTM gibi modeller, sıralı verilerin işlenmesinde ve zaman serileriyle ilgili görevlerde yaygın olarak kullanılmaktadır. Özellikle doğal dil işleme (NLP) alanında, dil modeli oluşturma ve metin sınıflandırma gibi görevlerde RNN ve LSTM'nin etkinliği dikkat çekmektedir. Lore ve arkadaşları tarafından yapılan bir çalışmada, RNN'nin dil modelleme ve konuşma tanıma gibi alanlarda başarılı sonuçlar verdiği gösterilmiştir (Lore ve ark., 2016)

RNN ve LSTM, sağlık alanında da önemli uygulama alanlarına sahiptir. Örneğin, hastane verileri üzerinde yapılan bir çalışmada, hastaların yaşam süresi tahminleri ve yoğun bakım ünitesindeki durumu gibi zaman serileri verilerine dayanan tahminlerde LSTM tabanlı modeller başarılı sonuçlar elde etmiştir (Rajkomar ve ark., 2018). Ayrıca, RNN tabanlı modeller, kalp ritmi ve EKG sinyalleri gibi biyomedikal verilerin analizi için de yaygın olarak kullanılmaktadır. Bu tür verilerin sınıflandırılması ve anormalliklerin tespit edilmesinde yüksek doğruluk sağlamaktadır (Kansal ve ark., 2023).

GAN modelleri, veri oluşturma, veri artırma ve stil transferi gibi görevlerde yeni olanaklar sunan modellerdir. Goodfellow ve arkadaşlarının yaptığı çalışma, GAN'lerin özellikle veri sentezi ve görüntü oluşturma görevlerinde başarılı olduğunu ortaya koymaktadır (Goodfellow, 2016). GAN'ler, tıp alanında da önemli uygulamalara sahiptir; örneğin, düşük çözünürlüklü tıbbi görüntülerin yüksek çözünürlüklü hale getirilmesi için GAN tabanlı yaklaşımlar kullanılmıştır. Bu tür bir uygulama, daha net ve doğru

teşhislerin yapılmasına olanak sağlamaktadır (Uzunova ve ark., 2020)

Ayrıca, GAN'lerin veri artırma amacıyla kullanıldığı çalışmalarda, eğitim veri setinin çeşitliliği artırılarak modelin genelleme yeteneği geliştirilmektedir. Özellikle sınırlı veri setleri ile çalışılan tıbbi alanlarda, GAN'lerin ürettiği sahte veriler sayesinde model performansı artırılabilir (Frid-Adar ve ark., 2018). Bunun dışında, GAN'ler, sanat ve eğlence sektöründe de kullanılmaktadır. Örneğin, stil transferi ve sahte sanat eserlerinin anlaşılmasında etkili çözümler sunmaktadır (Zhang ve ark., 2019).

ViT, görüntü işleme için Transformer tabanlı bir yaklaşım sunmakta ve özellikle büyük veri kümelerinde yüksek performans sergilemektedir. Karadeniz' in melanoma cilt kanseri görüntüleri üzerinden yapılan sınıflandırılma çalışmasında, ViT'nin transfer öğrenme ile birleştirildiğinde etkili sonuçlar verdiği gösterilmiştir (Karadeniz, 2024.). Bu tür yaklaşımlar, ViT'nin tıbbi görüntüleme gibi kritik alanlarda uygulanabilirliğini kanıtlamaktadır.

ViT, aynı zamanda endüstriyel görüntü işleme görevlerinde de kullanılmıştır. Özellikle üretim hatlarında kalite kontrol için kullanılan ViT tabanlı modeller, ürünlerin sınıflandırılması ve hata tespiti gibi görevlerde yaygın olarak uygulanmaktadır (Chen ve ark., 2021) Eğitim verisi büyük olduğunda ve çeşitliliği fazla olduğunda, ViT'nin yüksek doğruluk oranlarıyla sonuçlar verdiği görülmektedir.

Derin öğrenme mimarileri, özellikle tıbbi tanı modelleri geliştirme konusunda oldukça başarılı sonuçlar vermektedir. Cömert ve arkadaşları tarafından yapılan çalışmada, COVID-19 enfeksiyonunun tespit edilmesinde derin özelliklerin Bayes optimizasyonu ile birlikte kullanıldığı ve bu sayede teşhis doğruluğunun artırıldığı bildirilmiştir (Comert ve ark., 2020). Benzer bir şekilde, beyin tümörü teşhisi için yapılan çalışmalarda, farklı özellik çıkarma algoritmalarının seçilmesi ve CNN'in optimizasyon algoritmaları ile birlikte kullanılması, tanı modellerinin doğruluğunu artırmakta önemli bir rol oynamaktadır (Başaran, 2022).

Tıbbi teşhislerde ayrıca ViT ve CNN gibi farklı mimarilerin kombinasyonları da kullanılmaktadır. Bu tür hibrit modeller, çeşitli görüntüleme tekniklerinden elde edilen verilerin analiz edilmesi ve daha doğru tanıların konulması için yaygın olarak uygulanmaktadır. Örneğin, manyetik rezonans görüntüleme (MRI) ve bilgisayarlı tomografi (CT) verilerinin sınıflandırılması ve anormalliklerin tespitinde yüksek doğruluk elde etmektedir (Krishnapriya ve ark, 2023).

Gradyan inişi, Adam, RMSprop ve Nesterov Momentumu gibi optimizasyon yöntemleri, derin öğrenme modellerinin eğitiminde yaygın olarak kullanılmaktadır. Bu teknikler, öğrenme hızını uyarlayarak modelin daha hızlı ve doğru bir şekilde eğitilmesine yardımcı olmaktadır (Kingma, 2014).

Derin öğrenmenin uygulama alanları, görüntü işleme, doğal dil işleme, sağlık, finans, otomotiv, tarım ve enerji gibi birçok sektörü kapsamaktadır. Her alan için farklı mimarilerin ve tekniklerin kullanıldığı görülmektedir (Yang ve ark., 2019)

3. Derin öğrenme mimarileri

Derin öğrenmenin başarısı, kullanılan farklı sinir ağı mimarilerine dayanmaktadır. Bu bölümde, yaygın olarak kullanılan CNN, RNN, GAN ve ViT mimarileri ele alınmaktadır.

3.1. Evrimsel sinir ağları (CNN)

Evrimsel Sinir Ağları (CNN), özellikle görüntü işleme ve sınıflandırma gibi bilgisayarla görme (computer vision) alanlarında yaygın olarak kullanılan güçlü bir derin öğrenme mimarisidir. CNN'in başarısının temelinde, verilerdeki yerel özelliklerin etkin bir şekilde öğrenilmesini sağlayan evrim katmanları yer almaktadır. Bu mimari, görüntülerdeki nesnelerin kenar, köşe ve doku gibi temel özelliklerini öğrenmek için filtreler kullanarak, daha yüksek soyutlama düzeyinde temsil edilen özellikleri çıkarmaktadır (Zeebaree ve ark., 2020). Bu süreç, ağırlıkların optimize edilmesi ve kayıp fonksiyonlarının minimize edilmesi yoluyla gerçekleşmekte ve bu da modelin doğruluğunu arttırmaktadır.

CNN'in yapısında üç ana bileşen bulunmaktadır. Bu bileşenler evrişim katmanları, havuzlama (pooling) katmanları ve tam bağlantılı (fully connected) katmanlardır. Evrişim katmanları, giriş görüntüsü üzerinde kayan filtreler (kernels) aracılığıyla belirli özellikleri algılayarak yerel bağlantılar oluşturmaktadır. Bu filtreler, belirli bir boyutta (örneğin 3x3 veya 5x5) tanımlanmakta ve kayan pencere (sliding window) yöntemi ile görüntü üzerinde hareket ederek, görüntünün farklı bölümlerinden özellikler çıkarmaktadır. Evrişim işlemi, formül olarak aşağıda ki şekilde gösterilmiştir.

$$f(x, y) = \sum_i \sum_j I(i, j) \cdot K(x - i, y - j)$$

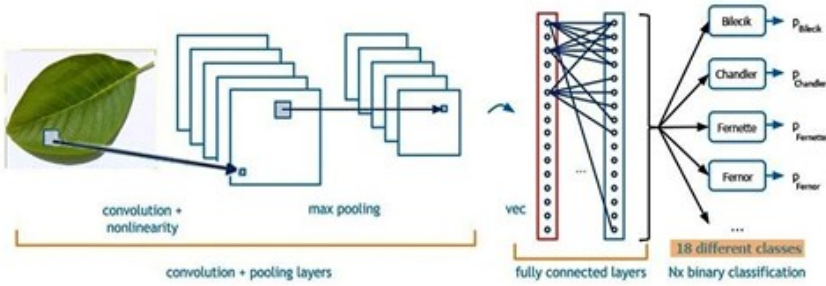
Bu formülde $I(i, j)$ giriş görüntüsündeki piksel değerini, K ise filtreyi ifade etmektedir. Toplam işlem sonucunda oluşan yeni matris, özellik haritası (feature map) olarak adlandırılmaktadır. Havuzlama katmanları, evrişim katmanları tarafından oluşturulan bu özellik haritalarının boyutunu küçültmek ve hesaplama maliyetini azaltmak amacıyla kullanılmaktadır. Bu katmanlar, genellikle maksimum havuzlama (max pooling) veya ortalama havuzlama (average pooling) yöntemleriyle çalışır. Örneğin, maksimum havuzlama 2x2'lik bir pencere üzerinde çalışarak, her bölgedeki en yüksek değeri seçmekte ve bu şekilde özellik haritasının boyutunu küçültmektedir.

Tam bağlantılı katmanlar ise ağın son aşamasında yer almakta ve sınıflandırma görevini gerçekleştirmektedir. Bu katmanlar, daha önceki katmanlarda öğrenilen özelliklerin bir araya getirilerek, belirli sınıflara ayrılmasını sağlamaktadır. Bu aşamada, kayıp fonksiyonu (loss function) olarak genellikle çapraz entropi (cross-entropy) veya kare hatalar (mean squared error) gibi yöntemler kullanılmakta ve modelin çıktı değerleri ile gerçek etiketler arasındaki fark minimize edilmektedir (Goodfellow, 2016). Aşağıdaki denklem, çapraz entropi kayıp fonksiyonunun nasıl hesaplandığını göstermektedir:

$$L = - \sum_i y_i \log(\hat{y}_i)$$

Burada y_i , gerçek etiketleri ve \hat{y}_i modelin tahmin ettiği olasılıkları ifade etmektedir. Bu formül, tahmin edilen ve gerçek etiketler arasındaki farkı ölçmekte ve bu farkın minimize edilmesi modelin doğruluğunu arttırmaktadır.

CNN genel mimarisi şekil 1’de gösterilmiştir (Karadeniz ve ark., 2023).



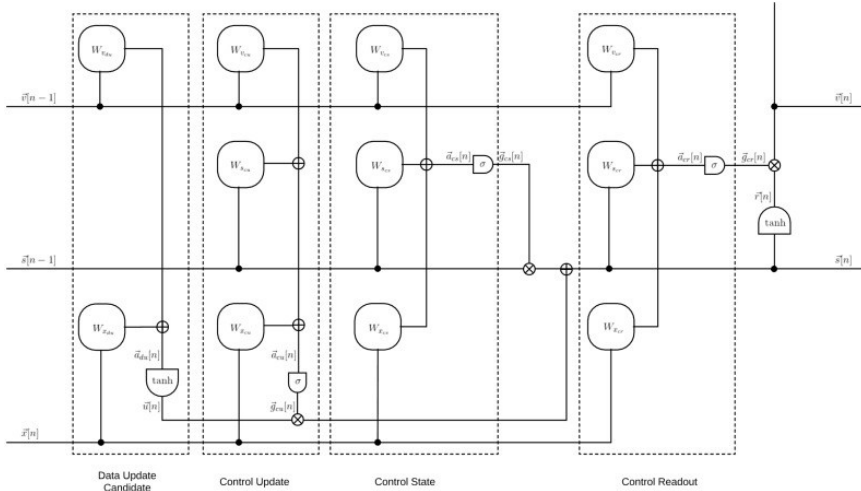
Şekil 1. CNN genel yapısı.

Sonuç olarak, CNN'ler, evrişim ve havuzlama katmanlarının sağladığı avantajlar sayesinde, özellikle görüntü işleme ve sınıflandırma gibi görevlerde yüksek doğrulukla çalışabilen derin öğrenme mimarileri arasında yer almaktadır. Özellikle büyük veri kümeleriyle eğitildiğinde ve uygun optimizasyon teknikleri kullanıldığında, bu ağlar nesne tanıma, tıbbi görüntü analizi ve endüstriyel kalite kontrol gibi birçok alanda etkili sonuçlar vermektedir (Bhatt ve ark., 2021).

3.2. Tekrarlayan sinir ağları (RNN) ve LSTM

Tekrarlayan Sinir Ağları (RNN), özellikle sıralı verilerin işlenmesi ve zaman serilerine dayalı uygulamalarda kullanılan güçlü bir derin öğrenme mimarisidir. RNN'ler, geçmiş zaman adımlarındaki bilgiyi hatırlayarak, bu bilgiyi sonraki adımların

tahmininde kullanma yeteneğine sahiptir. Bu özellik, dil modelleme, konuşma tanıma, zaman serisi tahmini ve biyomedikal sinyal işleme gibi alanlarda önemli bir avantaj sağlamaktadır (Lore ve ark., 2016). Ancak, RNN'lerin uzun süreli bağımlılıkları öğrenme konusunda bazı zorlukları bulunmaktadır. Bu sorun, özellikle gradyanların çok küçük veya çok büyük değerlere ulaşmasıyla ilgili olan gradyan kaybolma (vanishing gradient) ve patlama (exploding gradient) problemleri olarak bilinmektedir (Hochreiter, 1997). RNN temel mimarisi şekil 2'de gösterilmiştir (Sherstinsky, 2020).



Şekil 2. RNN genel mimarisi.

LSTM'ler, uzun süreli bağımlılıkları daha etkili bir şekilde öğrenmek amacıyla geliştirilen bir RNN türüdür. LSTM hücreleri, standart RNN'lerin aksine, bir "hücre durumu" (cell state) ve çeşitli kapılar (gates) kullanarak bilgiyi hatırlama veya unutma mekanizmasına sahiptir. Bu kapılar arasında unutma kapısı (forget gate), giriş kapısı (input gate) ve çıkış kapısı (output gate) bulunmaktadır. Bu yapılar, hücre durumunun güncellenmesi ve saklanacak bilginin belirlenmesi için birlikte çalışmaktadır (Hochreiter, 1997). Son yıllarda, Transformer tabanlı modeller, özellikle doğal dil işleme ve konuşma tanıma gibi görevlerde RNN

ve LSTM'lere kıyasla daha yüksek performans elde etmektedir. Transformer'lar, paralel işlem yapma yetenekleri ve dikkat mekanizması (attention mechanism) sayesinde uzun dizileri daha verimli bir şekilde işleyebilmektedir. Transformer modelleri, özellikle konuşma uygulamalarında RNN tabanlı yaklaşımlar ile karşılaştırıldığında daha düşük hata oranları sağlamaktadır (Vaswani ve ark., 2017). Bununla birlikte, RNN ve LSTM tabanlı modellerin, özellikle zaman serileri ve biyomedikal sinyal işleme gibi belirli görevlerde hala önemli avantajları bulunmaktadır. RNN ve LSTM tabanlı modeller, biyomedikal veri analizi, doğal dil işleme, konuşma tanıma ve zaman serisi tahmini gibi birçok alanda yaygın olarak kullanılmaktadır. LSTM hücrelerinin uzun vadeli bağımlılıkları öğrenme kapasitesi, özellikle dil modelleme ve biyomedikal sinyal işleme gibi alanlarda önemli bir avantaj sunmaktadır (Lipton, 2015). Örneğin, LSTM tabanlı modeller, kalp ritmi analizi, EKG sinyali sınıflandırma ve fonem tanıma gibi tıbbi uygulamalarda yüksek doğruluk oranlarıyla öne çıkmaktadır (Wang ve ark., 2019).

Sonuç olarak, RNN ve LSTM ağları, sıralı verilerin işlenmesinde kritik öneme sahip olmakla birlikte, yeni mimari ve optimizasyon tekniklerinin entegrasyonu ile daha geniş bir uygulama alanına hitap etmektedir. Bu ağlar, farklı mimarilerin avantajlarını bir araya getiren hibrit yaklaşımlar ve Transformer gibi yenilikçi yöntemlerle birlikte kullanılmaya devam etmektedir.

3.3. Üretici karşıt ağlar (GAN)

Üretici Karşıt Ağlar (Generative Adversarial Networks - GAN), derin öğrenme alanında önemli bir yenilik olarak, veri üretme ve öğrenme süreçlerini iyileştiren bir yaklaşım sunmaktadır. GAN'lerin temelinde, jeneratör (generator) ve ayırmacı (discriminator) olmak üzere iki farklı sinir ağı yer almaktadır. Bu ağlar karşılıklı bir rekabet içinde çalışmaktadır; jeneratör sahte veri üretirken, ayırmacı bu verilerin gerçek mi sahte mi olduğunu ayırt etmeye çalışır. Bu rekabetçi süreç, jeneratörün daha gerçekçi veriler üretmek için sürekli olarak kendisini geliştirmesini sağlamaktadır.

GAN'lerin bu rekabetçi yapısı, veri oluşturma, stil transferi, görüntü iyileştirme ve daha birçok alanda başarılı sonuçlar elde edilmesine olanak tanımaktadır (Goodfellow, 2016).

GAN'ler, jeneratör ve ayırmacı isimli iki ana bileşenden oluşmaktadır. Jeneratör, rastgele bir girdi vektöründen başlayarak sahte veri üretmekte ve bu sahte veriler, gerçek veriyle benzer özelliklere sahip olacak şekilde jeneratör tarafından öğrenilmektedir. Ayırmacı ise, sahte verileri gerçek veri ile karşılaştırarak bunların gerçek olup olmadığını belirlemeye çalışmaktadır. Eğitim sürecinde bu iki ağ, karşılıklı olarak iyileşmektedir. Jeneratör daha gerçekçi sahte veriler üretmeyi öğrenirken, ayırmacı da bu sahte verileri ayırt etmeye çalışmaktadır. GAN'lerin eğitim süreci, oyun teorisi perspektifinde bir minimaks problem olarak ele alınabilir; yani jeneratör, ayırmacıyı kandırmak için çalışırken, ayırmacı ise en doğru sınıflandırmayı yapmak için eğitilmektedir (Goodfellow, 2016).

GAN'ler, çeşitli uygulama alanlarında yaygın olarak kullanılmaktadır. Bu uygulamalar arasında görüntü oluşturma, stil transferi, veri artırma, görüntü iyileştirme ve süper çözünürlük gibi görevler yer almaktadır. Örneğin, stil transferinde, bir görüntüdeki stilin başka bir görüntüye uygulanması mümkündür ve bu, sanat eserlerinin dijital olarak dönüştürülmesi veya fotoğraflara çeşitli stil efektlerinin eklenmesi gibi alanlarda kullanılmaktadır (Isola ve ark., 2017).

GAN'lerin farklı varyantları da mevcuttur ve bunlar, çeşitli uygulama ihtiyaçlarına yönelik olarak geliştirilmiştir:

- Koşullu GAN (cGAN): Bu tür GAN'ler, jeneratör ve ayırmacıyı belirli bir koşul veya etiket verilerek eğitmektedir, bu da belirli türde verilerin üretilmesini sağlamaktadır (Mirza, 2014).
- Derin Konvolüsyonlu GAN (DCGAN): Bu model, jeneratör ve ayırmacının derin konvolüsyonel katmanlar kullanarak daha karmaşık yapılar öğrenmesine olanak tanımaktadır (Radford, 2015).

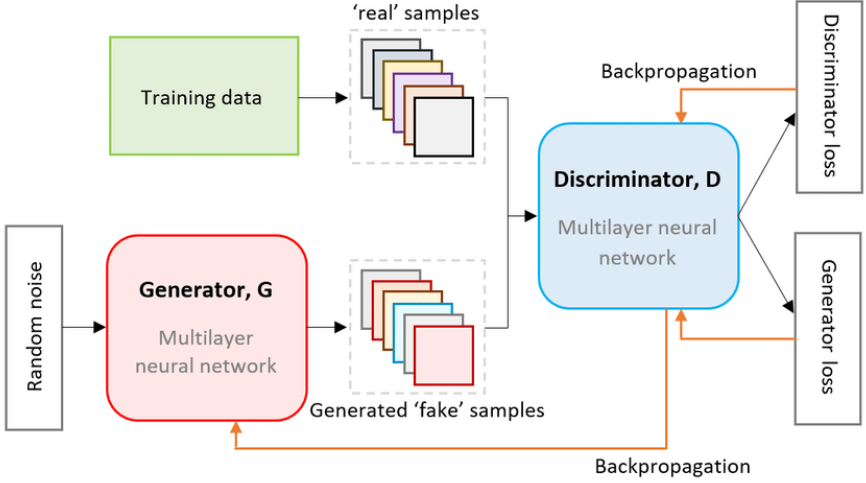
- CycleGAN: İki farklı veri kümesi arasında dönüşüm yapabilen bu model, görüntülerin stil değişimi gibi görevlerde oldukça başarılı sonuçlar vermiştir.. Örneğin, bir yaz manzarasını kış manzarasına dönüştürmek veya siyah-beyaz bir fotoğrafı renklendirmek için kullanılabilir (Zhu ve ark., 2017).

GAN'lerin avantajları dışında bazı kısıtlamaları da mevcuttur. Eğitim sürecindeki dengesizlik, jeneratör ve ayırmıcının eşzamanlı olarak öğrenememesine neden olabilmektedir. Bu durumda jeneratör, aşırı başarılı olduğunda ayırmıcının yaptığı sınıflandırma anlamsız hale gelmekte veya tam tersi, ayırmıcı çok güçlü olduğunda jeneratör iyi bir şekilde öğrenememektedir (Goodfellow, 2016).

Bu tür kısıtları aşmak için çeşitli çözümler geliştirilmiştir:

- Wasserstein GAN (WGAN): Geleneksel GAN'lerdeki dengesizlik sorununu, farklı bir mesafe ölçütü kullanarak daha kararlı bir eğitim süreci sağlayan bir yöntemle çözmeyi amaçlamaktadır (Arjovsky ve ark., 2017)
- Geçici Öğrenme Oranı (Adaptive Learning Rate): Eğitim sürecinde hem jeneratör hem de ayırmıcı için dinamik öğrenme oranları kullanarak, dengesizliklerin önüne geçebilmektedir (Gulrajani ve ark., 2017).

Sonuç olarak, GAN'ler, yaratıcı ve gerçekçi veri üretme yeteneği ile büyük bir potansiyele sahiptir ve birçok yenilikçi uygulama sunmaktadır. Jeneratör ve ayırmıcı arasındaki rekabetçi yapı, özellikle veri oluşturma ve dönüşüm gibi alanlarda dikkate değer çözümler üretmektedir. GAN mimarisinin genel yapısı şekil 3'te gösterilmiştir (Little ve ark., 2021).



Şekil 3. GAN mimarisi.

3.4. Vision transformer (ViT)

Vision Transformer (ViT), son yıllarda bilgisayarla görme alanında önemli başarılar elde etmiş görüntü işleme için Transformer tabanlı bir yaklaşım sunan, popüler bir derin öğrenme modelidir. Geleneksel olarak, görüntü işleme görevlerinde yaygın olarak kullanılan Evrişimsel Sinir Ağları'ndan (CNN) farklı olarak, ViT, tamamen Transformer tabanlı bir yaklaşım benimser ve bu sayede yüksek paralellik kapasitesi sunar. ViT'nin temel çalışma prensibi, görüntüleri küçük parçalara (patch) ayırarak, her bir parçayı düzleştirip, bir diziye dönüştürmek ve bu parçaların karşılıklı ilişkilerini öğrenmektir (Karadeniz, 2024).

ViT'nin yapısında, ilk olarak giriş görüntüsü sabit boyutlu küçük parçalara bölünmektedir (örneğin, 16x16 veya 32x32 piksellik parçalar). Her bir parça, düzleştirilerek bir lineer katman aracılığıyla sabit boyutlu bir vektöre dönüştürülmektedir. Bu vektörler, daha sonra pozisyon bilgisi eklenerek (position embedding) modelin girişine sunulmaktadır. Pozisyon bilgisi, parçalar arasındaki konumsal ilişkileri korumak için kullanılmakta

ve bu sayede görüntüdeki mekânsal düzenin öğrenilmesine yardımcı olmaktadır (Dosovitskiy, 2020).

ViT'nin mimarisinin temeli, çoklu dikkat mekanizmasına (multi-head attention mechanism) dayanan Transformer bloklarından oluşmaktadır. Bu bloklar, görüntü parçaları arasındaki ilişkileri öğrenmek için dikkat mekanizmasını kullanmaktadır. Paralel işlem yapabilme yeteneği sayesinde büyük veri kümelerinde yüksek performans göstermektedir. Bu yapı, her bir görüntü parçasının diğer parçalarla olan ilişkisini öğrenmek için kullanarak, görüntünün genel anlamını kavrayabilmektedir.

ViT'nin temel avantajlarından biri, paralel işlem yapabilme yeteneği sayesinde büyük veri kümeleri üzerinde daha verimli bir şekilde çalışabilmesidir. Geleneksel CNN tabanlı yöntemler, evrişim katmanları ile yerel özellikleri öğrenirken, ViT, tüm görüntü üzerinde dikkat mekanizmasını kullanarak daha geniş kapsamlı ilişkileri öğrenebilmektedir. CNN'ler yerel örüntüleri tanımakta üstün olsalar da, ViT'nin geniş kapsamlı öğrenme yeteneği, görüntüdeki daha karmaşık desenlerin daha iyi anlaşılmasını sağlamaktadır (Khan ve ark., 2022).

ViT'nin yüksek performans göstermesi için büyük miktarda veriye ihtiyaç vardır. CNN tabanlı modeller, nispeten daha az veri ile etkili sonuçlar elde edebilmektedir. ViT önceden eğitilmiş büyük veri kümelerinden transfer öğrenme teknikleri uygulayarak, daha küçük veri setlerinde de etkili sonuçlar elde edebilmektedir (Touvron ve ark., 2021).

ViT, görüntü sınıflandırma, nesne algılama, görüntü segmentasyonu ve tıbbi görüntü analizi gibi birçok bilgisayarla görme uygulamasında etkili sonuçlar elde etmiştir. Örneğin, ViT'nin cilt kanseri tespiti gibi tıbbi görüntüleme görevlerinde yüksek doğruluk sağladığı gösterilmiştir (Karadeniz, 2024). Bu tür uygulamalarda ViT, görüntüdeki karmaşık desenleri ve özellikleri tanıyarak doğru teşhis yapılmasına olanak tanımaktadır.

ViT ayrıca metin tabanlı veriler ile birlikte görüntü verilerini birleştirerek çoklu modelli öğrenme (multimodal learning) görevlerinde kullanılabilir. Örneğin, görüntü ve metin verilerini birlikte analiz eden çalışmada, ViT'nin kullanımı, görsel soru yanıtı (visual question answering) ve görüntü açıklama (image captioning) gibi görevlerde başarı sağlamaktadır (Maier ve ark., 2021).

ViT'nin başarısının ardından, çeşitli ViT varyantları ve iyileştirilmiş ViT modelleri geliştirilmiştir. Bu modeller, ViT'nin temel prensiplerini kullanarak daha karmaşık görevlerde daha iyi performans elde etmeyi amaçlamaktadır.

- DeiT (Data-efficient Image Transformer): ViT'nin veri gereksinimini azaltmak için tasarlanmış bir varyanttır. DeiT, bilgi aktarımı ve veri artırma tekniklerini kullanarak daha az veri ile etkili sonuçlar elde etmektedir (Touvron ve ark., 2021).

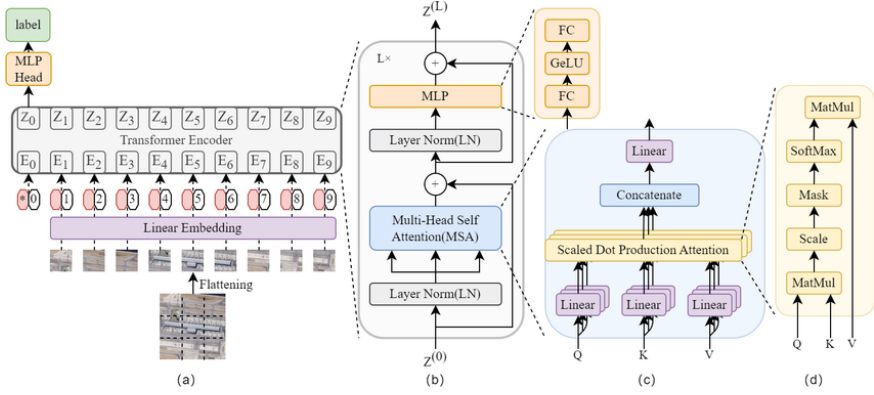
- Swin Transformer: Yerel ve küresel dikkat mekanizmalarını birleştirerek, daha küçük veri setlerinde doğruluğu geliştirmektedir (Liu ve ark., 2021).

- Hybrid ViT: Hem CNN hem de ViT'nin avantajlarını birleştirir. Bu mimari, evrimsel katmanları kullanarak düşük seviyeli özellikleri çıkarırken, ViT blokları ile daha yüksek seviyeli ilişkileri öğrenmektedir (Abramo ve ark., 2021)

Transfer öğrenme, veri artırma teknikleri ve düşük kaynaklı ortamlarda daha verimli modeller geliştirilmesi gibi alanlarda çalışmalar devam etmektedir (Dosovitskiy, 2020)

Sonuç olarak, Vision Transformer (ViT), görüntü işleme ve bilgisayarla görme alanlarında önemli bir yenilik sunan bir derin öğrenme mimarisi olarak öne çıkmaktadır. Transformer tabanlı yapısı sayesinde büyük veri kümelerinde yüksek performans gösteren ViT, gelecekteki görsel veri işleme teknolojilerinde de önemli bir rol oynamaya devam edecektir.

ViT temel mimarisi şekil 4'te gösterilmiştir (Bi ve ark., 2022).



Şekil 4. ViT mimarisi.

4. Eğitim süreçleri ve aşırı öğrenmeyi önleme teknikleri

Derin öğrenme modellerinin etkinliği, eğitim süreçlerinin doğru bir şekilde uygulanması ve aşırı öğrenmenin önlenmesiyle sağlanmaktadır.

4.1. Optimizasyon teknikleri

- Gradyan İnişi (Gradient Descent): Modelin hatasını azaltmak için kullanılan temel bir algoritmadır.
- Adam: Gradyan inişinin uyarlanabilir bir versiyonu olup, dinamik öğrenme oranı sağlamaktadır.
- RMSprop: Öğrenme hızını adapte eder ve konveks olmayan kayıp fonksiyonlarında iyi sonuçlar vermektedir.
- Nesterov Momentumu: Gradyan yönünün gelecekteki tahminini kullanarak eğitim sürecini hızlandırmaktadır (Bartlett ve ark., 2021).

4.2. Aşırı öğrenmenin önlenmesi

- Dropout: Eğitim sırasında bazı nöronların rastgele kapatılması aşırı uyumu önlemektedir. (Srivastava ve ark., 2014)
- Düzenleme (Regularization): L1 ve L2 düzenleme yöntemleri, modelin aşırı karmaşık hale gelmesini engellemektedir (Li ve ark., 2023).

5. Derin öğrenmenin uygulama alanları

Derin öğrenme, geniş uygulama alanı sayesinde farklı sektörlerde önemli dönüşümler yaratmıştır. Çeşitli veri türleriyle çalışabilme ve karmaşık ilişkileri öğrenme yeteneği, bu teknolojiyi birçok alanda vazgeçilmez hale getirmiştir. Bu bölümde, derin öğrenmenin önemli uygulama alanları ayrıntılı olarak ele alınmaktadır.

5.1. Görüntü İşleme

Görüntü işleme, derin öğrenmenin en yaygın kullanıldığı alanlardan biridir. Derin öğrenme algoritmaları, nesne tanıma, görüntü sınıflandırma ve segmentasyon gibi görevlerde oldukça yüksek doğruluk oranları sunmaktadır. Evrimsel Sinir Ağları (CNN) gibi mimariler, tıbbi görüntülerde hastalık teşhisi, güvenlik kameraları ile anormal olayların tespiti ve otonom araçlarda çevre algılama gibi çeşitli görüntü işleme uygulamalarında önemli başarılar elde etmiştir. Bu teknolojinin tıbbi görüntülerde, özellikle kanser teşhisi gibi karmaşık ve kritik görevlerde kullanımı, hastalıkların daha erken ve daha doğru bir şekilde tespit edilmesi, görüntülerden sınıflandırma yapılması sağlamaktadır (Esteva ve ark., 2019). Karadeniz, Çelik & Başaran., 2023).

Endüstriyel kalite kontrol uygulamalarında da derin öğrenme tabanlı görüntü işleme çözümleri yaygın olarak kullanılmaktadır. Üretim hatlarında ürünlerin görsel analizi, hatalı veya kusurlu ürünlerin otomatik olarak tespit edilmesi ve üretim süreçlerinin optimize edilmesi bu çözümler sayesinde mümkün olmaktadır.

Ayrıca, yapay zeka destekli görüntü işleme teknikleri ile tarım sektöründe bitki hastalıklarının erken tespit edilmesi ve tarım arazilerinin izlenmesi gibi görevler de başarıyla gerçekleştirilmektedir (Zeebaree ve ark., 2020).

5.2. Doğal Dil İşleme (NLP)

Doğal dil işleme (NLP), dil modelleri oluşturma, metin sınıflandırma, duygu analizi ve makine çevirisi gibi görevlerde derin öğrenmenin yaygın olarak kullanıldığı bir diğer önemli alandır. Özellikle, Transformer tabanlı modeller ve RNN/LSTM gibi sıralı verilerle çalışan ağlar, insan dilinin karmaşık yapısını anlamada ve işleme görevlerinde yüksek performans göstermektedir. Dil modelleri oluşturma, metin tabanlı bilgi çıkarma ve diyalog sistemleri gibi uygulamalarda, derin öğrenme tabanlı NLP çözümleri geleneksel yöntemlere kıyasla daha doğru ve doğal sonuçlar üretmektedir (Yang ve ark., 2019)

Ayrıca, derin öğrenme tabanlı NLP teknikleri, sağlık alanında elektronik sağlık kayıtlarının analizi, tıbbi metin madenciliği ve klinik dokümantasyonun otomatikleştirilmesi gibi görevlerde de kullanılmaktadır. Bu tür uygulamalar, büyük miktarda metin verisinin işlenmesi ve analiz edilmesi gereken durumlarda, doktorların ve araştırmacıların iş yükünü önemli ölçüde azaltabilir ve daha verimli sağlık hizmeti sunulmasını sağlayabilir (Ferrari & Esuli, 2019).

5.3. Sağlık ve Tıp

Sağlık ve tıp alanında derin öğrenme, biyomedikal sinyallerin işlenmesi, genetik veri analizi, tıbbi görüntülerin otomatik analizi ve hastalık teşhisi gibi çeşitli görevlerde kullanılmaktadır. Örneğin, radyolojik görüntülerde kanserli hücrelerin tespiti, biyomedikal sinyallerde anormal kalp ritmi analizleri ve genetik veri setlerinde hastalık risklerinin tahmin edilmesi gibi alanlarda derin öğrenme teknikleri yüksek doğruluk oranları sunmaktadır (Esteva ve ark., 2019). Bu tür uygulamalar,

sağlık hizmetlerinin daha kişiselleştirilmiş ve hızlı bir şekilde sunulmasını mümkün kılmaktadır.

Ayrıca, ilaç keşfi süreçlerinde de derin öğrenme algoritmaları önemli rol oynamaktadır. Kimyasal bileşiklerin biyolojik aktivitesini tahmin etme, potansiyel ilaç adaylarını değerlendirme ve klinik deney sonuçlarını analiz etme gibi görevlerde kullanılan derin öğrenme tabanlı modeller, yeni ilaçların keşfedilme süresini kısaltabilir ve araştırma maliyetlerini azaltabilmektedir (Mittal & Hasija, 2020).

5.4. Finans ve Ekonomi

Finans ve ekonomi alanında derin öğrenme, borsa, kredi risk analizi, dolandırıcılık tespiti ve müşteri davranışlarının analizi gibi çeşitli görevlerde yaygın olarak kullanılmaktadır. Bu teknolojinin kullanımı, finansal işlemlerdeki karmaşık ilişkileri ve örüntüleri tanımlamayı kolaylaştırarak daha doğru tahminler yapılmasını sağlar. Derin öğrenme tabanlı modeller, özellikle borsa tahminlerinde büyük veri setleri üzerinde çalışarak gelecekteki piyasa hareketlerini tahmin etmekte ve bu sayede yatırımcılara avantaj sağlamaktadır (Heaton ve ark., 2017). Kredi risk analizinde ise, müşterilerin geçmiş kredi bilgileri ve diğer demografik verilerine dayanarak kredi başvurularının onaylanma olasılığını hesaplayan derin öğrenme algoritmaları kullanılmaktadır. Ayrıca, dolandırıcılık tespiti uygulamalarında, anormal işlemleri ve dolandırıcılık faaliyetlerini belirlemek için derin öğrenme tabanlı anomali algılama yöntemleri tercih edilmektedir. Bu uygulamalar, finansal güvenliği artırarak kurumların kayıplarını minimize etmektedir.

5.5. Oyun Geliştirme ve Yapay Zeka

Oyun geliştirme alanında, yapay zeka karakterlerinin daha doğal ve insana benzer davranışlar sergilemesi için derin öğrenme algoritmalarından yararlanılmaktadır. Derin takviyeli öğrenme (deep reinforcement learning), strateji oyunlarında yapay zeka oyuncularının insan oyuncularla rekabet edebilecek seviyede

öğrenmesini sağlamaktadır (Z. Chen & Huang, 2017). Bu yöntem, karmaşık oyun ortamlarında dinamik ve adaptif davranışlar sergileyen karakterler geliştirmek için kullanılabilir.

Ayrıca, oyunlardaki grafik ve görüntü iyileştirme işlemlerinde de derin öğrenme kullanılmaktadır. Örneğin, oyun grafiklerini gerçekçi hale getiren süper çözünürlük (super-resolution) ve stil transferi gibi görüntü işleme teknikleri ile oyun deneyimi artırılabilir. Derin öğrenme algoritmalarının oyun geliştirmede kullanılmasının en büyük avantajı, oyunların daha sürükleyici ve gerçekçi bir hale getirilmesidir.

5.6. Otomotiv ve Otonom Araçlar

Otomotiv sektörü ve otonom araç teknolojileri, derin öğrenmenin önemli uygulama alanlarından biridir. Otonom araçlar, çevresel algılama, nesne tespiti ve rota planlaması için derin öğrenme tekniklerini kullanmaktadır. Özellikle, otonom sürüş sistemlerinde kullanılan CNN ve ViT tabanlı görüntü işleme algoritmaları, araçların trafikteki nesnelere tanınmasını ve güvenli bir şekilde yönlendirilmesini sağlar (Grigorescu ve ark., 2020).

Ayrıca, sürücü yardım sistemleri (ADAS) gibi uygulamalarda da derin öğrenme teknikleri yaygın olarak kullanılmaktadır. Bu sistemler, şerit takip, çarpışma öncesi uyarı ve kör nokta tespiti gibi görevleri yerine getirerek sürüş güvenliğini artırmaktadır. Derin öğrenme tabanlı algılama ve karar alma sistemleri, otonom araç teknolojisinin daha güvenli ve etkin bir şekilde geliştirilmesine katkı sağlamaktadır.

5.7. Tarım ve Gıda Teknolojisi

Tarım alanında derin öğrenme, ürün hastalıklarının teşhisi, verimlilik artırma, mahsul tahmini ve zararlı tespiti gibi çeşitli görevlerde yaygın olarak kullanılmaktadır (Karadeniz et al., 2024). Özellikle, bitki hastalıklarının erken tespiti için kullanılan derin öğrenme tabanlı görüntü işleme teknikleri, çiftçilere mahsullerini koruma ve ürün kaybını önleme konusunda yardımcı olmaktadır (Kamilaris ve Prenafeta-Boldú, 2018).

Gıda teknolojisinde ise derin öğrenme, gıda kalitesini değerlendirme, paketlenme hatlarındaki kalite kontrol süreçleri ve gıda işleme süreçlerinin optimize edilmesi gibi alanlarda kullanılmaktadır. Bu teknikler, üretim sürecinde hatalı ürünlerin tespit edilmesini ve gıda güvenliğinin artırılmasını sağlar. Tarım ve gıda sektöründeki bu uygulamalar, verimliliği artırırken maliyetlerin düşürülmesine katkıda bulunmaktadır.

5.8. Enerji ve İklim Bilimleri

Derin öğrenme uygulamaları, enerji ve iklim bilimleri alanında, enerji üretimi ve tüketimini optimize etmek, hava durumu tahminlerini geliştirmek ve iklim değişikliği analizleri yapmak için kullanılmaktadır. Örneğin, enerji talebini tahmin etme ve yenilenebilir enerji kaynaklarının verimliliğini artırma gibi görevlerde derin öğrenme tabanlı modeller yüksek doğruluk oranları sunmaktadır (Reichstein ve ark., 2019).

İklim bilimleri alanında ise, atmosferik verilerin analizi, deniz seviyelerindeki değişikliklerin tahmini ve iklim modellerinin iyileştirilmesi gibi görevlerde derin öğrenme tekniklerinden yararlanılmaktadır. Bu uygulamalar, çevresel sorunlara yönelik daha doğru ve hızlı çözümler sunarak, sürdürülebilirlik ve doğal kaynakların yönetimi konularında önemli katkılar sağlamaktadır.

6. Tartışma

Derin öğrenme, çok katmanlı yapılar sayesinde büyük ve karmaşık veri kümelerinden anlamlı özellikler çıkarmada başarılıdır. Özellikle görüntü işleme ve doğal dil işleme gibi alanlarda geleneksel yöntemlere kıyasla daha yüksek doğruluk sağlamaktadır (LeCun ve ark., 2015).

Derin öğrenmenin en önemli sınırlamaları arasında yüksek hesaplama maliyetleri, büyük veri gereksinimi ve açıklanabilirlik sorunları bulunmaktadır. Bu sınırlamalar, özellikle veri sınırlı ortamlarda veya hesaplama kaynaklarının sınırlı olduğu durumlarda önemli zorluklar çıkarabilir (Schmidhuber, 2015).

Federatif öğrenme, açıklanabilir yapay zeka ve kuantum hesaplama gibi alanlarda yapılan arařtırmalar, derin öğrenmenin sınırlamalarını aşmak için umut vadetmektedir. Ayrıca, daha az veri ile eğitim sağlayan tekniklerin geliştirilmesi ve modelin şeffaflığını artıran çözümler de önemli araştırma alanlarıdır (Garg & Ramakrishnan, 2020).

Tablo 1. Derin öğrenme mimarileri karşılaştırma tablosu.

Mimari	Avantajlar	Dezavantajlar	Uygulama Alanları
CNN (Evrışimsel Sinir Ağları)	Görüntü işleme ve sınıflandırmada yüksek doğruluk sağlar.	Uzun sıralı verilerle başa çıkmakta zorlanır.	Görüntü tanıma, tıbbi görüntü analizi, nesne tanıma.
RNN (Tekrarlayan Sinir Ağları)	Sıralı verilerle çalışma yeteneği ve uzun bağımlılıkları öğrenebilme.	Uzun süreli bağımlılıklarda gradyan kaybolma sorunları yaşanabilir.	Doğal dil işleme, ses tanıma, dil modeli oluşturma.
GAN (Generative Adversarial Networks)	Gerçekçi veri üretimi ve veri artırmada etkilidir.	Eğitim süreci hassastır ve dengesiz olabilir (jeneratör ve ayırmacı dengesizliği).	Görüntü oluşturma, stil transferi, veri sentezi.
ViT (Vision Transformer)	Paralel işleme yeteneği sayesinde büyük veri kümelerinde iyi performans gösterir.	Hesaplama gereksinimi yüksektir ve daha fazla veri gerektirir.	Görüntü işleme, doğal dil işleme, tıbbi görüntü sınıflandırması.

7. Sonuç

Derin öğrenme, yapay zekanın en hızlı gelişen ve geniş uygulama alanına sahip teknolojilerinden biri olarak kabul edilmektedir. Bu alan, çok katmanlı sinir ağlarının karmaşık verileri işleme ve anlamlandırma yeteneği sayesinde görüntü işleme, doğal dil işleme, sağlık, finans, otomotiv, tarım ve enerji gibi birçok sektörde önemli başarılarla imza atmıştır (LeCun ve ark., 2015). Derin öğrenmenin bu geniş uygulama yelpazesi, çeşitli veri türleri ve problem alanları için optimize edilmiş farklı mimarilerin, yani CNN, RNN, GAN ve ViT gibi yapılarının kullanılmasıyla mümkün olmuştur. Bu mimarilerin her biri, kendi avantajları ve sınırlamaları ile belirli görevlerde yüksek performans sağlamaktadır.

Bu gelişmelerin yanı sıra, derin öğrenmenin sınırlamaları ve karşılaşılan zorluklar da dikkate alınmalıdır. Bu sınırlamalar arasında en yaygın olanları yüksek hesaplama maliyetleri, büyük veri gereksinimi ve modelin iç işleyişinin açıklanabilirliği ile ilgili sorunlardır (Schmidhuber, 2015). Derin öğrenme modellerinin genellikle büyük ölçekli veri kümeleri üzerinde daha yüksek doğruluk sağladığı görülse de, küçük veri setleriyle etkili sonuçlar elde etmek zordur. Bu durum, özellikle sağlık ve biyomedikal gibi veri sınırlı alanlarda ciddi bir engel oluşturabilir. Ayrıca, modelin açıklanabilirlik sorunları, özellikle kritik kararların verildiği tıp ve güvenlik gibi alanlarda önemli bir endişe kaynağıdır.

Bu zorlukların üstesinden gelmek için, derin öğrenme araştırmaları federatif öğrenme, açıklanabilir yapay zeka (Explainable AI) ve kuantum hesaplama gibi yeni yaklaşımlara yönelmektedir. Federatif öğrenme, kullanıcı verilerini korurken merkezi olmayan bir şekilde model eğitimi sağlamaktadır. Özellikle sağlık ve finans gibi hassas veri gerektiren alanlarda önemli bir gelişme olarak değerlendirilmektedir (Yang ve ark., 2019). Açıklanabilir yapay zeka ise, modellerin karar mekanizmalarını daha anlaşılır hale getirmeyi amaçlamakta ve bu sayede kritik alanlarda güvenilir yapay zeka uygulamalarının geliştirilmesine olanak tanımaktadır.

Derin öğrenmenin daha fazla benimsenmesi ve daha geniş çapta uygulanabilmesi için, mevcut sınırlamaların üstesinden gelmek amacıyla yeni yöntemler ve teknikler geliştirilmelidir. Bu çerçevede, hibrit modellerin kullanımı, yani CNN ve ViT gibi farklı mimarilerin bir arada kullanılması, farklı veri türlerinin işlenmesi ve daha karmaşık problemlerin çözülmesi için etkili bir yaklaşım olarak öne çıkmaktadır (Khan ve ark., 2022). Aynı şekilde, veri artırma teknikleri ve transfer öğrenme gibi yöntemler, daha küçük veri setlerinde de yüksek performans elde edilmesini sağlamak için kullanılabilir.

Sonuç olarak, derin öğrenme alanındaki gelişmeler, hem akademik hem de endüstriyel uygulamalarda büyük bir etki ile devam etmektedir. Bu teknoloji, çeşitli veri türlerini işleyebilme ve karmaşık sorunları çözme kabiliyeti sayesinde birçok sektörde başarı ile kullanılmaya devam etmektedir. Gelecekte, derin öğrenmenin daha güvenli, daha hızlı ve daha erişilebilir çözümler sunması beklenmektedir. Bu ilerlemeler, insan yaşamını daha verimli hale getiren yapay zeka çözümleri sunarak, birçok farklı alanda kullanılma potansiyeline sahiptir.

Kaynaklar

Abramo, G., Aksnes, D. W., & D'Angelo, C. A. (2021). Gender differences in research performance within and between countries: Italy vs Norway. *Journal of Informetrics*, *15*(2), 101144.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan, arXiv. *ArXiv Preprint ArXiv:1701.07875*.

Bartlett, P. L., Montanari, A., & Rakhlin, A. (2021). Deep learning: a statistical viewpoint. *Acta Numerica*, *30*, 87–201.

Başaran, E. (2022a). A new brain tumor diagnostic model: Selection of textural feature extraction algorithms and convolution neural network features with optimization algorithms. *Computers in Biology and Medicine*, *148*, 105857.

Başaran, E. (2022b). A new brain tumor diagnostic model: Selection of textural feature extraction algorithms and convolution neural network features with optimization algorithms. *Computers in Biology and Medicine*, *148*, 105857.

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., & Ghayvat, H. (2021). CNN variants for computer vision: History, architecture, application, challenges and future scope. *Electronics*, *10*(20), 2470.

Bi, M., Wang, M., Li, Z., & Hong, D. (2022). Vision transformer with contrastive learning for remote sensing image scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *16*, 738–749.

Buckner, C. (2019). Deep learning: A philosophical introduction. *Philosophy Compass*, *14*(10), e12625.

Chen, J., He, Y., Frey, E. C., Li, Y., & Du, Y. (2021). Vit-v-net: Vision transformer for unsupervised volumetric medical image registration. *ArXiv Preprint ArXiv:2104.06468*.

Chen, Z., & Huang, X. (2017). End-to-end learning for lane keeping of self-driving cars. *2017 IEEE Intelligent Vehicles Symposium (IV)*, 1856–1860.

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

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., & Dean, J. (2019). A guide to deep learning in healthcare. *Nature Medicine*, 25(1), 24–29.

Ferrari, A., & Esuli, A. (2019). An NLP approach for cross-domain ambiguity detection in requirements engineering. *Automated Software Engineering*, 26(3), 559–598.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, 321, 321–331.

Garg, S., & Ramakrishnan, G. (2020). Advances in quantum deep learning: An overview. *ArXiv Preprint ArXiv:2005.04316*.

Goodfellow, I. (2016). *Deep learning*. MIT press.

Grigorescu, S., Trasnea, B., Cocias, T., & Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3), 362–386.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in Neural Information Processing Systems*, 30.

Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1), 3–12.

Hochreiter, S. (1997). Long Short-term Memory. *Neural Computation MIT-Press*.

Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1125–1134.

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). A review of the use of convolutional neural networks in agriculture. *The Journal of Agricultural Science*, 156(3), 312–322.

Kansal, S., Garg, D., Upadhyay, A., Mittal, S., & Talwar, G. S. (2023). DL-AMPUT-EEG: Design and development of the low-cost prosthesis for rehabilitation of upper limb amputees using deep-learning-based techniques. *Engineering Applications of Artificial Intelligence*, 126, 106990.

Karadeniz, A. T. (n.d.-a). 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. (n.d.-b). 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., & Celik, 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.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 54(10s), 1–41.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980*.

Krishnapriya, S., & Karuna, Y. (2023). Pre-trained deep learning models for brain MRI image classification. *Frontiers in Human Neuroscience*, 17, 1150120.

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

Li, Z., Gao, E., Zhou, J., Han, W., Xu, X., & Gao, X. (2023). Applications of deep learning in understanding gene regulation. *Cell Reports Methods*, 3(1).

Lipton, Z. C. (2015). Learning to diagnose with LSTM recurrent neural networks. *ArXiv Preprint ArXiv:1511.03677*.

Little, C., Elliot, M., Allmendinger, R., & Samani, S. S. (2021). Generative adversarial networks for synthetic data generation: a comparative study. *ArXiv Preprint ArXiv:2112.01925*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

Lore, K. G., Stoecklein, D., Davies, M., Ganapathysubramanian, B., & Sarkar, S. (2016). Deep action sequence learning for causal shape transformation. *ArXiv Preprint ArXiv:1605.05368*.

Maier, J., Öhlinger, D., Schmid, U., Függer, M., & Nowak, T. (2021). A composable glitch-aware delay model. *Proceedings of the 2021 on Great Lakes Symposium on VLSI*, 147–154.

Mirza, M. (2014). Conditional generative adversarial nets. *ArXiv Preprint ArXiv:1411.1784*.

Mittal, S., & Hasija, Y. (2020). RNA Secondary Structure Prediction using Machine Learning: A Review. *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, 714–719.

Nour, M., Cömert, Z., & Polat, K. (2020). A novel medical diagnosis model for COVID-19 infection detection based on deep features and Bayesian optimization. *Applied Soft Computing*, *97*, 106580.

Radford, A. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *ArXiv Preprint ArXiv:1511.06434*.

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., Liu, P. J., Liu, X., Marcus, J., & Sun, M. (2018). Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, *1*(1), 1–10.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., & Prabhat, F. (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, *566*(7743), 195–204.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117.

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers &

distillation through attention. *International Conference on Machine Learning*, 10347–10357.

Uzunova, H., Ehrhardt, J., & Handels, H. (2020). Memory-efficient GAN-based domain translation of high resolution 3D medical images. *Computerized Medical Imaging and Graphics*, 86, 101801.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need.(Nips), 2017. *ArXiv Preprint ArXiv:1706.03762*, 10, S0140525X16001837.

Wang, J., Zhang, Y., Ma, Q., Huang, H., & Hong, X. (2019). Deep learning for single-channel EEG signals sleep stage scoring based on frequency domain representation. *International Conference on Health Information Science*, 121–133.

Yang, H., Luo, L., Chueng, L. P., Ling, D., & Chin, F. (2019). Deep learning and its applications to natural language processing. *Deep Learning: Fundamentals, Theory and Applications*, 89–109.

Zeebaree, S. R., Ahmed, O., & Obid, K. (2020). Csaernet: An efficient deep learning architecture for image classification. *2020 3rd International Conference on Engineering Technology and Its Applications (IICETA)*, 122–127.

Zhang, T.-F., Tilke, P., Dupont, E., Zhu, L.-C., Liang, L., & Bailey, W. (2019). Generating geologically realistic 3D reservoir facies models using deep learning of sedimentary architecture with generative adversarial networks. *Petroleum Science*, 16(3), 541–549.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2223–2232.

BÖLÜM VII

Güncel metasezgisel optimizasyon algoritmalarının CEC2020 test fonksiyonları ile değerlendirilmesi

Rabia UYAR¹

Giriş

Son yıllarda, doğadan esinlenen algoritmalar, karmaşık optimizasyon problemlerinin çözümü için etkin bir alternatif olarak öne çıkmaktadır (Dumlu & Yavuz, t.y.; Sadollah vd., 2015). Bu algoritmalar, doğadaki canlıların arama, avlanma, çiftleşme ve hayatta kalma stratejilerinden ilham alınarak geliştirilmiştir (Yagmur vd., 2024). Parçacık sürü optimizasyonu (PSO), karınca koloni optimizasyonu (ACO) ve yapay arı kolonisi (ABC) gibi algoritmalar bu yaklaşımın önde gelen örneklerindedir (Alatas, 2019; Altay, 2022). Bu bağlamda, son dönemde geliştirilen Dwarf Mongoose optimizasyon algoritması (DMO) ve Nutcracker

¹ Arş Gör., Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Kütahya/TÜRKİYE, Orcid: 0000-0002-6565-0560, rabia.uyar@dpu.edu.tr

optimizasyon (NOA) algoritmaları, performanslarıyla dikkat çekmektedir (Abdel-Basset vd., 2023; Agushaka vd., 2022).

DMO, cüce firavun faresinin topluluk içindeki avlanma ve sosyal etkileşim davranışlarını taklit ederek çalışan bir optimizasyon algoritmasıdır. Firavun farelerinin avı takip etme, gruplarının hareketlerini koordine etme ve bireylerin kolektif zekâsını kullanma yetenekleri, algoritmanın temel işleyişini oluşturmaktadır. Bu yaklaşım, arama uzayını daha etkin bir şekilde taramak ve küresel çözümü daha hızlı bulmak için kullanılır (Gunawan & Kristian, 2023; Özen Kavas vd., 2023; Yavuz, 2022a). NOA ise, fındıkkıran kuşlarının besin döngüsü ve besin zincirindeki bireylerin etkileşimlerinden ilham alarak geliştirilmiş bir algoritmadır. NOA, bireylerin besin kaynaklarını tespit etme, beslenme ve enerji transferi süreçlerini modelleyerek optimizasyon için yeni bir bakış açısı sunmaktadır. Bu algoritma, arama uzayının hem yerel hem de küresel çözümlerini daha verimli bir şekilde araştırabilmek için besin arama stratejilerini simüle etmektedir. Her iki algoritmanın özelliklerini öne çıkaran çalışmalar incelendiğinde, bu algoritmaların belirli türdeki optimizasyon problemlerinde diğerlerinden daha öne çıktığı görülmektedir (Yavuz, 2022b). DMO ve NOA, doğadan esinlenen diğer optimizasyon algoritmaları ile karşılaştırıldığında, daha düşük hesaplama maliyetleri ve daha iyi küresel arama kapasitesi sunmaktadır (Arslan vd., 2024; Dörterler vd., 2024). Ancak bu algoritmaların performanslarının sistematik bir şekilde karşılaştırılması gerekmektedir (Aydemir, 2022; Dursun & Serttaş, 2024).

Performans değerlendirmesi için CEC 2020 ölçüt fonksiyonları önemli bir rol oynamaktadır (Bai vd., 2023; Eroz &

Tanyildizi, 2018; Houssein vd., 2023). CEC (Congress on Evolutionary Computation) tarafından belirlenen bu ölçüt fonksiyonları, optimizasyon algoritmalarının etkinliğini değerlendirmek için yaygın olarak kullanılmaktadır. Bu fonksiyonlar, karmaşık çok boyutlu uzaylarda çok sayıda yerel çözümün bulunduğu zorlu arama problemlerini temsil etmektedir. Bu nedenle, CEC 2020 test fonksiyonları, DMO ve NOA gibi yeni algoritmaların gücünü ölçmek için standart bir kıyaslama platformu sunmaktadır.

Çalışmanın amacı, DMO ve NOA algoritmalarının CEC 2020 ölçüt fonksiyonları üzerindeki performanslarının karşılaştırılmasıdır. Bu kapsamda, her iki algoritmanın standart sapma ve ortalama değerleri incelenecek ve başarımları karşılaştırılacaktır.

MATERYAL VE METHOD

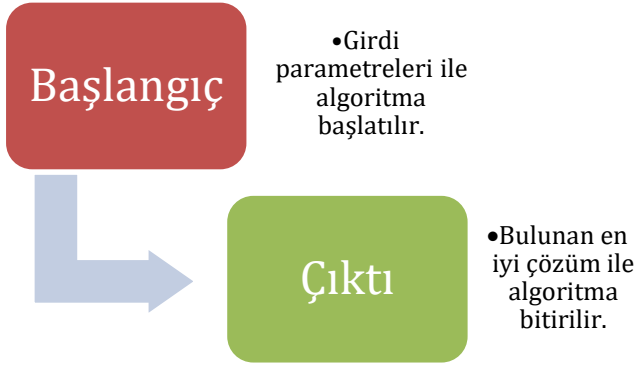
1. Nutcracker optimizasyon algoritması (NOA)

Nutcracker optimizasyon algoritması, doğadan esinlenen bir meta-sezgisel optimizasyon algoritmasıdır. Bu algoritma, fındıkkıran kuşlarının doğadaki yiyecek arama ve depolama davranışlarını taklit ederek geliştirilmiştir. Bu kuşların doğal davranışları, optimizasyon problemlerinde çözümleri arama ve bulma sürecine uyarlanmıştır. Özellikle küresel optimizasyon problemlerinde, arama alanındaki en iyi çözümleri keşfetmek için hem keşif (exploration) hem de sömürü (exploitation) aşamalarına odaklanmaktadır.

NOA algoritmasının başarısını belirleyen en önemli unsurlardan biri, keşif ve sömürü dengesini etkili bir şekilde

kurabilmesidir. Algoritma, çözüm arama sürecini iki temel aşamada gerçekleştirir. Keşif aşamasında, arama alanının daha az bilinen bölümleri araştırılarak potansiyel iyi çözümlerin tespiti sağlanır. Bu aşama, çözüm uzayının geniş bir kısmını kapsayarak çeşitliliği artırmayı hedefler. Ardından gelen sömürü aşamasında ise, önceden keşfedilen iyi çözümler üzerinde yoğunlaşılır. Bu aşamada, daha iyi çözümlere ulaşmak için mevcut çözümlerin çevresi ayrıntılı bir şekilde incelenir. Bu iki aşama arasındaki dengenin iyi kurulması, algoritmanın genel performansı açısından kritik bir rol oynamaktadır.

Bununla birlikte, NOA algoritması, diğer birçok doğadan esinlenen algoritmada olduğu gibi, yerel optima sıkışması sorunuyla karşılaşabilmektedir. Özellikle karmaşık ve çok modlu optimizasyon problemlerinde, algoritma en iyi çözümü bulamadan, arama alanındaki yerel bir optimum noktaya saplanıp kalabilir. Bu durum, hem keşif hem de sömürü aşamalarının etkinliğini doğrudan etkiler. Yerel optima sıkışmasını önlemek için literatürde çeşitli stratejiler geliştirilmiş olup, bu stratejiler genellikle keşif kapasitesini artırmaya yönelik iyileştirmeleri içermektedir (Bai vd., 2023; Balochian & Baloochian, 2019; Eroz & Tanyildizi, 2018; Li vd., 2020). Aşağıda yer alan şekil 1'de Fındıkkıran optimizasyon algoritmasının akış diyagramı sunulmaktadır.



Şekil 1 NOA'nın akış diyagramı

Şekil 1'de sunulan akış diyagramı incelendiğinde, algoritmanın girdi parametreleri ile başladığı ve en iyi çözüm bulunduğu ise sonlandığı görülmektedir.

1.1.Nutcracker optimizasyon algoritmasının sözde kodu

Tablo 1'de Fındıkkıran optimizasyon algoritmasının başlangıcı için gerekli olan girdi parametreleri ve açıklamaları sunulmaktadır.

Tablo 1. NOA'nın başlangıç girdi parametreleri

Girdi Parametreleri	Açıklama
N	Popülasyon boyutu
L	Değişkenlerin alt sınırları
U	Değişkenlerin üst sınırları
t=0	Mevcut iterasyon sayısı
T _{max}	Maksimum iterasyon sayısı

Burada tablo 1'de sunulan N parametresi, algoritmanın çözüm arama sürecinde kullanılacak olan popülasyon boyutunu ifade

ederek çözüm adaylarının sayısını belirlemektedir. L ve U parametreleri, sırasıyla değişkenlerin alt ve üst sınırlarını tanımlayarak arama uzayının kapsamını sınırlandırır. $t=0$ ifadesi, algoritmanın başlangıç anındaki mevcut iterasyon sayısını temsil etmektedir. T_{\max} parametresi ise algoritmanın sonlandırılacağı maksimum iterasyon sayısını belirleyerek döngünün durma koşulunu ortaya koymaktadır.

Şekil 2’de Fındıkkıran optimizasyon algoritmasının sözde kodu sunulmaktadır.

1. N adet fındıkkıran/çözüm rastgele olarak oluşturulur. (Denklem 1 ile)
2. Her çözümün uygunluk değeri hesaplanır ve popülasyondaki en iyi birey belirlenir.
3. $t=1$ olarak ayarlanır.
4. Doğru olduğu sürece; $t < T_{\max}$
 - ✓ Rastgele σ ve σ_1 (0 ile 1 arasında) sayıları üretilir.
 - ✓ Eğer $\sigma < \sigma_1$:
 - (Yiycek toplama ve depolama stratejisi uygulanır)
 - Φ rastgele bir sayı olarak 0 ile 1 arasında belirlenir.
 - Her $i=1: N$:
 - Her $j=1: d$
 - ❖ Eğer $\phi > P_a$:
 - Keşif Aşaması 1 uygulanır.
 - X_i^{t+1} değeri Denklem (2 ve 3) kullanılarak güncellenir.
 - ❖ Aksi takdirde:
 - Sömürü aşaması 1 uygulanır.
 - X_i^{t+1} Denklem (4 ve 5) kullanılarak güncellenir.
 - SON
 - $t, t = t+1$ olarak güncellenir.
 - ✓ SON
 - ✓ Aksi takdirde:
 - (Cache-Arama ve Kurtarma Stratejisi uygulanır)
 - RP matrisi denklemler (6,7 ve 8) ile oluşturulur.
 - Rastgele bir ϕ sayısı (0 ile 1 arasında) üretilir.
 - Her $i=1: N$:

- Eğer $\phi > Pa$:
 - Keşif Aşaması 2 uygulanır.
 - X_i^{t+1} Denklemler (9,10 ve 11) ile güncellenir.
- Aksi takdirde:
 - Sömürü Aşaması 2 uygulanır.
 - X_i^{t+1} Denklem (12,13 ve 14) ile güncellenir.
- $t=t+1$ olarak güncellenir.
- ✓ SON
- 5. SON
- 6. En iyi çözümü döndür.

Şekil 2. NOA algoritmasının sözde kodu

NOA'nın sözde kodu incelendiğinde, algoritma N adet rastgele çözüm oluşturulmasıyla başlamaktadır ve her çözümün uygunluk değeri hesaplanarak popülasyonun en iyi bireyi belirlenmektedir. İterasyon sayısı $t=1$ olarak ayarlandıktan sonra, $t < T_{max}$ koşulu sağlandığı sürece algoritma çalışmaktadır. İlk olarak, rastgele σ ve σ_1 değerleri üretilir ve $\sigma < \sigma_1$ koşuluna göre strateji seçimi yapılmaktadır. Eğer bu koşul sağlanırsa, Yiyecek Toplama ve Depolama Stratejisi uygulanır; burada Keşif Aşaması 1 veya Sömürü Aşaması 1 seçilmekte ve çözüm, ilgili denklemler kullanılarak güncellenmektedir. Aksi durumda, Cache-Arama ve Kurtarma Stratejisi uygulanmaktadır; bu aşamada Keşif Aşaması 2 veya Sömürü Aşaması 2 seçilmekte ve çözüm yine ilgili denklemlerle güncellenmektedir. Her iterasyon sonunda t değeri güncellenir. Döngü tamamlandığında algoritma, en iyi çözümü döndürerek sonlanmaktadır. Bu yapı, keşif ve sömürü stratejilerini dengeleyerek optimizasyon sürecini gerçekleştiren sistematik bir çerçeveye sunmaktadır.

1.2.Nutcracker optimizasyon algoritmasında kullanılan denklemler

Bu bölümde, fındikkıran optimizasyon algoritmasının tüm aşamalarında kullanılan denklemler ve parametrelerin açıklamaları sunulmaktadır.

- $\vec{X}_{i,j}^t$: i'inci çözümün j'inci değişkeninin t iterasyonundaki değeridir.
- \vec{U}_j : j'inci değişkenin üst sınırıdır.
- \vec{L}_j : j'inci değişkenin alt sınırıdır.
- RM: Rastgele bir çarpan veya faktör, genellikle optimizasyonu yönlendiren bir parametredir.
- $\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6$: Belirli geçiş şartlarını belirleyen rastgele sayılardır. Bu parametreler genellikle keşif ve sömürü aşamaları arasında geçişi kontrol etmektedir.
- $\gamma, \mu, \alpha, \lambda$: Adaptasyon parametreleri olup, çözümün nasıl güncelleneceğini belirleyen çarpanlardır. Genellikle çözümün değişim hızını veya yönünü ayarlamak için kullanılmaktadır.
- r_1, r_2 : Rastgele sayılar olup, genellikle çözümün rastgeleliğini veya çeşitliliğini arttırmak için kullanılmaktadır.
- ϕ : Yine bir rastgele sayı olup, beslenme stratejilerinin uygulanmasında kullanılmaktadır.
- \vec{X}_{best}^t : t iterasyonundaki en iyi çözümün değeridir.
- RP: Cache veya bellek tabanlı stratejinin kullanıldığı durumlarda, geçmiş çözümleri içeren matrisi temsil etmektedir. Bu matris, çözüm arama sürecinde önceki çözümlerle ilişki kurmayı sağlamaktadır.
- $f(\vec{X}_i^t)$: i'inci çözümün t iterasyonundaki uygunluk değeri, yani çözümün kalitesini değerlendiren fonksiyondur.

- X_A^t, X_B^t : A ve B noktalarındaki çözüm değerleri, genellikle farklı popülasyon üyelerinin çözüm değerlerini temsil etmektedir.

$$1. \vec{X}_{i,j}^t = +(\vec{U}_j - \vec{L}_j) \cdot \overline{RM} + \vec{L}_j \quad \text{Burada; } l = 1, 2, \dots, N, \quad j = 1, 2, \dots, D$$

$$2. \begin{cases} X_{i,j}^t & \text{if } \tau_1 < \tau_2 \\ X_{m,j}^t + \gamma \cdot (X_{Aj}^t - X_{Bj}^t) + \mu \cdot (r^2 \cdot U_j - L_j) & \text{if } t \frac{T_{max}}{2.0}, \\ & \text{Aksi takdirde,} \\ X_{C,j}^t + \mu \cdot (X_{Aj}^t - X_{Bj}^t) + \mu \cdot (r_1 < \delta)(r^2 \cdot U_j - L_j) & \text{, Aksi takdirde} \end{cases}$$

$$3. \vec{X}_i^{t+1} = \begin{cases} \vec{X}_i^{t+1}, & \text{if } (\vec{X}_i^{t+1} < f(\vec{X}_i^t)) \\ \vec{X}_i^t, & \text{Aksi takdirde} \end{cases}$$

$$4. \vec{X}_i^{t+1(new)} \begin{cases} \vec{X}_i^t + \mu \cdot (\vec{X}_{best}^t - \vec{X}_i^t) \cdot |\lambda| + r_1 \cdot (X_A^t - X_B^t) & \text{if } \tau_1 < \tau_2 \\ \vec{X}_{best}^t + \mu \cdot (X_A^t - X_B^t) & \text{if } \tau_1 < \tau_2 \\ \vec{X}_{best}^t \cdot l & \text{Aksi takdirde} \end{cases}$$

$$5. \vec{X}_i^{t+1} = \begin{cases} \vec{X}_i^{t+1}, & \text{if } (\vec{X}_i^{t+1} < f(\vec{X}_i^t)) \\ \vec{X}_i^t, & \text{Aksi takdirde} \end{cases}$$

$$6. RPS = \begin{bmatrix} \overrightarrow{RP}_{1,1}^t & \overrightarrow{RP}_{1,2}^t \\ \vdots & \vdots \\ \overrightarrow{RP}_{i,1}^t & \overrightarrow{RP}_{i,2}^t \\ \vdots & \vdots \\ \overrightarrow{RP}_{N,1}^t & \overrightarrow{RP}_{N,2}^t \\ \vdots & \vdots \end{bmatrix}$$

$$7. \overrightarrow{RP}_{i,1}^t = \begin{cases} \vec{X}_i^t + \alpha \cdot \cos(\theta) \cdot ((\vec{X}_A^t - \vec{X}_B^t)) + \alpha \cdot RP, & \text{if } \theta = \pi/2 \\ \vec{X}_i^t + \alpha \cdot \cos(\theta) \cdot ((\vec{X}_A^t - \vec{X}_B^t)), & \text{Aksi takdirde} \end{cases}$$

$$8. \overrightarrow{RP}_{i,2}^t = \begin{cases} (\vec{X}_i^t + \alpha \cdot \cos(\theta) \cdot ((\vec{U} - \vec{L}) \cdot \tau_3 \overrightarrow{L}) + \alpha \cdot RP) \cdot \vec{U}_2, & \text{if } \theta = \pi/2 \\ \vec{X}_i^t + \alpha \cdot \cos(\theta) \cdot ((\vec{U} - \vec{L}) \cdot \tau_3 \overrightarrow{L}) \cdot \vec{U}_2, & \text{Aksi takdirde} \end{cases}$$

$$9. X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t, & \text{if } \tau_3 < \tau_4 \\ X_{i,j}^t + r_1 \cdot (X_{best.j}^t - X_{i,j}^t) + r_2 \cdot (\overrightarrow{RP}_{i,1}^t - X_{C,j}^t), & \text{Aksi takdirde} \end{cases}$$

$$10. X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t, & \text{if } \tau_5 < \tau_6 \\ X_{i,j}^t + r_1 \cdot (X_{best.j}^t - X_{i,j}^t) + r_2 \cdot (\overrightarrow{RP}_{i,1}^t - X_{C,j}^t), & \text{Aksi takdirde} \end{cases}$$

$$11. \vec{X}_i^{t+1} = \begin{cases} \vec{X}_i^{t+1}, & \text{if } (\vec{X}_i^{t+1} < f(\vec{X}_i^t)) \\ \vec{X}_i^t, & \text{Aksi takdirde} \end{cases}$$

$$12. \vec{X}_i^{t+1} = \begin{cases} \vec{X}_i^t, & \text{if } f(\vec{X}_i^t) < \text{if } (\overline{RP}_{i,1}^t) \\ \overline{RP}_{i,1}^t, & \text{Aksi takdirde} \end{cases}$$

$$13. \vec{X}_i^{t+1} = \begin{cases} \vec{X}_i^t, & \text{if } f(\vec{X}_i^t) < \text{if } (\overline{RP}_{i,2}^t) \\ \overline{RP}_{i,2}^t, & \text{Aksi takdirde} \end{cases}$$

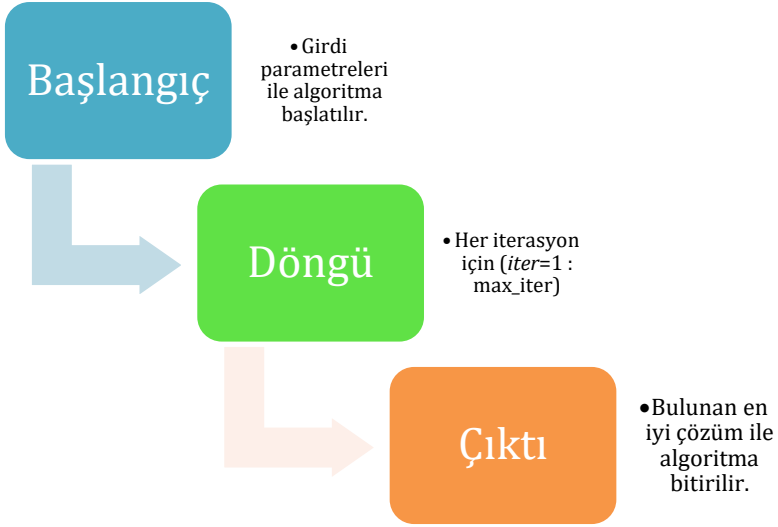
$$14. \vec{X}_i^{t+1} = \begin{cases} \vec{X}_i^{t+1}, & \text{if } (\vec{X}_i^{t+1} < f(\vec{X}_i^t)) \\ \vec{X}_i^t, & \text{Aksi takdirde} \end{cases}$$

2. Dwarf Mongoose optimizasyon algoritması (DMO)

DMO, küresel optimizasyon problemlerinin çözümüne yönelik olarak, cüce firavun farelerinin avlanma ve sosyal etkileşim davranışlarını modelleyen ve bu davranışları algoritmik olarak gerçekleyen bir yaklaşımdır. Cüce firavun faresi optimizasyon algoritması, doğadaki cüce firavun faresi sürüsünün avlanma davranışlarından esinlenerek geliştirilmiş yenilikçi bir optimizasyon yöntemidir. Algoritmanın ilk adımında, başlangıç aşaması yer alır. Bu aşamada, cüce firavun faresi sürüsünün bireyleri, arama uzayına

rastgele bir şekilde yerleştirilir. Her bir birey, olası çözümleri temsil eden bir konuma sahiptir.

Algoritmanın temel aşamalarından biri, keşif ve saldırı sürecidir. Bu aşamada fareler, avın yerini tespit ettikçe konumlarını güncellemektedirler. Daha iyi çözümler (yani daha avantajlı avlar) keşfedildiğinde, tüm sürü bu çözüme doğru yönelir. Bu, bireysel keşif aşamasından kolektif bir harekete geçişi simgelemektedir. Köşeye sıkıştırma aşamasında ise, fareler, avı yakalamak için bireysel pozisyonlarını optimize eder. Bu süreç, arama uzayını daraltarak daha iyi çözümlere ulaşmayı sağlamaktadır (Faramarzi vd., 2020). Daraltma işlemi, aramanın yerel bir bölgeye odaklanmasını sağlamakta ve sömürü aşamasını güçlendirmektedir. Bir diğer önemli aşama ise sosyal iletişim sürecidir. Fareler, sürüdeki diğer bireylerin konumlarını göz önünde bulundurarak konumlarını günceller. Bu ise, bireysel keşif ile kolektif arama dengesini kurarak sürünün genel performansını artırmaktadır. Her bir birey, sürünün diğer üyelerinin bilgilerini dikkate alarak hareket etmektedir. Bu etkileşim, algoritmanın keşif ve sömürü dengesini kurmasında önemli bir rol oynamaktadır (Yavuz vd., 2024). Tüm bu aşamalar, döngüsel bir süreç olarak belirli bir durdurma kriteri karşılanana kadar devam etmektedir. Durdurma kriteri genellikle maksimum iterasyon sayısı veya en iyi çözümde bir değişiklik olmaması gibi bir ölçüte dayanmaktadır. Bu döngü boyunca, sürü üyeleri hem bireysel hem de kolektif arama süreçlerini etkili bir şekilde yöneterek, global optimum çözüme ulaşmaya çalışmaktadır (Mirjalili & Lewis, 2016).



Şekil 3 DMO'nun akış diyagramı

2.1. Dwarf Mongoose optimizasyon algoritmasının sözde kodu

Tablo 2’de Cüce firavun faresi optimizasyon algoritmasının başlangıç için gerekli olan girdi parametreleri ve açıklamaları sunulmaktadır.

Tablo 2. DMO'nun başlangıç girdi parametreleri

Girdi Parametreleri	Açıklama
peep	Beslenme davranışı
n	Arama ajanları
bs	Bakıcı sayısı
L	Bakıcı değişim parametresi
C	Zaman sayacını

Burada tablo 2’de sunulan DMO algoritmasının başlangıç girdi parametreleri, algoritmanın performansını doğrudan etkileyen temel bileşenlerdir. Peep parametresi, cüce firavun farelerinin beslenme davranışını modelleyerek, arama ajanlarının keşif (exploration) ve sömürü (exploitation) dengesi üzerinde etkili olmaktadır. n parametresi, optimizasyon sürecine katılan arama ajanlarının (fare bireylerinin) sayısını belirler ve bu sayının artırılması, çözüm uzayının daha geniş bir bölümünün keşfedilmesini sağlamaktadır. bs parametresi, doğadaki bakıcı (caretaker) bireylerin rolünü yansıtarak, bilgi paylaşımı ve görev dağılımını kontrol etmektedir. L parametresi, ajanlar arasındaki görev değişimini ifade eder ve görev değişiklikleri ile algoritmanın çeşitliliğini artırarak yerel minimumlara sıkışmayı önlemektedir. Son olarak, C parametresi, algoritmanın çalışması sırasında geçen süreyi veya iterasyon sayısını takip eden bir sayaç olarak görev yapar ve genellikle durdurma kriteri olarak kullanılmaktadır. Bu parametreler birlikte, DMO algoritmasının keşif-sömürü dengesi, görev değişimi ve hesaplama maliyetleri açısından etkin bir şekilde çalışmasını sağlamaktadır.

Şekil 4’te Cüce firavun faresi optimizasyon algoritmasının sözde kodu sunulmaktadır.

1. Denklem göre α 'yı hesapla (Denklem 15 ile).
2. Yeni bir yiyecek pozisyonu adayı üret.
3. Burada ϕ , rastgele bir sayı olarak alınır.
4. X_{i+1} için yeni uygunluk değerini değerlendir (Denklem 16 ile).
5. Uyku tepesi değerlerini hesapla (Denklem 17 ile).
6. Tüm uyku tepelerinin ortalamasını hesapla (Denklem 18 ile).

7. Hareket vektörünü hesapla (Denklem 19 ile)
8. Eğer $C \geq L$, bakıcı değişimini uygula ve bakıcıların pozisyonlarını başlat.
9. Uygunluk değerlerini hesapla.
10. Eğer $fit_i \leq \alpha$, izci farenin sonraki pozisyonunu simüle et (Denklem 20 ile).
11. Şimdiye kadar bulunan en iyi çözümü güncelle.

Son Döngü

12. En iyi çözümü döndür.

Şekil 4. DMO algoritmasının sözde kodu

DMO'nun sözde kodu incelendiğinde, cüce firavun farelerinin avlanma ve sosyal etkileşim davranışlarını modelleyen bir süreç olarak tasarlanmaktadır. İlk adımda, α parametresi Denklem 15'e göre hesaplanır ve bu parametre, uygunluk eşiği olarak görev yapmaktadır. Ardından, rastgele bir sayı olan ϕ kullanılarak yeni bir yiyecek pozisyonu adayı üretilmektedir. Üçüncü adımda, bu yeni pozisyonun uygunluk değeri Denklem 16'ya göre değerlendirilmektedir. Devamında, ajanların uyku tepesine (sleep mound) ilişkin değerleri Denklem 17 ile hesaplanmakta ve tüm uyku tepelerinin ortalaması Denklem 18 ile bulunmaktadır. Bu ortalama, bir sonraki adımda hareket vektörünün belirlenmesinde kullanılmaktadır (Denklem 19). Eğer $C \geq L$ koşulu sağlanırsa, bakıcı değişimi gerçekleşmekte ve bakıcıların pozisyonları yeniden başlatılarak uygunluk değerleri hesaplanmaktadır. Bu aşamada, uygunluk değerleri α 'dan küçük olan ajanlar için izci farelerin sonraki pozisyonları Denklem 20'ye göre simüle edilmektedir. Tüm bu süreç boyunca, şimdiye kadar elde edilen en iyi çözüm

güncellenir. Son döngü tamamlandığında, algoritma en iyi çözümü döndürerek süreci sonlandırılmaktadır. Bu adımların her biri, arama sürecinin keşif (exploration) ve sömürü (exploitation) dengesini sağlamak, görev değişimi ile çeşitliliği artırmak ve küresel optimuma ulaşmayı kolaylaştırmak amacıyla tasarlanmıştır.

2.2. Dwarf Mongoose optimizasyon algoritmasında kullanılan denklemler

$$15. \alpha = \frac{fit_i}{\sum_{i=1}^n fit_i}$$

$$16. X_{i+1} = X_i + \phi \cdot peep$$

$$17. sm_i = \frac{fit_{i,t+1} - fit_{i,t}}{\max(|fit_{i,t+1}|, |fit_{i,t}|)}$$

$$18. \phi = \frac{\sum_{i=1}^n sm_i}{n}$$

$$19. \vec{M} = \sum_{i=1}^n \frac{X_i \cdot sm_i}{X_i}$$

$$20. X_{i+1} = \begin{cases} X_i - CF \cdot rand. (X_i - \vec{M}) & \text{Eğer } \phi_{i+1} > \phi_i \quad (\text{Keşif}) \\ X_i + CF \cdot rand. (X_i - \vec{M}) & \text{Diğer durum} \quad (\text{Sömürü}) \end{cases}$$

3.CEC 2020 Ölçüt setinde kullanılan fonksiyonlar ve özellikleri

CEC 2020 test fonksiyonları, optimizasyon algoritmalarının performansını değerlendirmek ve karşılaştırmak için geniş ölçekte kullanılan standartlaştırılmış bir test setidir. Bu fonksiyonlar, farklı

yapılar ve zorluk derecelerine sahip problemler içererek, algoritmaların hem keşif (exploration) hem de sömürü (exploitation) yeteneklerini test etmeyi amaçlamaktadır. Fonksiyonlar genellikle dört ana kategoriye ayrılır: Unimodal fonksiyonlar, multimodal fonksiyonlar, hibrit fonksiyonlar ve kompozisyon fonksiyonları. Aşağıda, CEC 2020 test fonksiyonlarının kategorileri ve özellikleri detaylandırılmaktadır.

3.1. Unimodal fonksiyonlar

Unimodal fonksiyonlar, yalnızca bir küresel minimuma sahip olan fonksiyonlardır. Bu tür fonksiyonlar, algoritmaların küresel bir optimuma hızlı bir şekilde ulaşma yeteneğini değerlendirmek için tasarlanmıştır.

- Shifted and Rotated Bent Cigar Fonksiyonu (F1): Bu fonksiyon, optimizasyon problemlerinde basit yapıların modellenmesini sağlamaktadır. Çözüm uzayında tek yönlü bir eğim göstererek algoritmaların doğrusal olmayan problemlerdeki başarımını test etmek için kullanılmaktadır. Optimal Değer (F_i): 100

3.2. Multimodal fonksiyonlar

Multimodal fonksiyonlar, birden fazla yerel minimum içermesiyle karakterize edilmektedir. Bu fonksiyonlar, algoritmaların yerel minimumlardan kaçma ve küresel optimumu bulma yeteneklerini ölçmek için kullanılmaktadır.

- Shifted and Rotated Schwefel Fonksiyonu (F2): Bu fonksiyon, çok sayıda yerel minimum içerir ve algoritmaların keşif yeteneğini değerlendirmektedir. Optimal Değer (F_i): 1100

- Shifted and Rotated Lunacek bi-Rastrigin Fonksiyonu (F3): İki çekim merkezi ile karmaşık bir yapı sunmaktadır ve algoritmaların farklı alanları keşfetmesini gerektirmektedir. Optimal Değer (Fi): 700
- Expanded Rosenbrock's Plus Griewangk's Fonksiyonu (F4): İki farklı matematiksel fonksiyonun birleşimi olan bu yapı, algoritmaların zorlu eğim yapılarında nasıl performans gösterdiğini test etmektedir. Optimal Değer (Fi): 1900

3.3. Hibrit fonksiyonlar

Hibrit fonksiyonlar, birden fazla temel fonksiyonun birleşiminden oluşmaktadır ve algoritmaların karmaşık problem yapılarına nasıl adapte olabildiğini değerlendirir.

- Hibrit Fonksiyon 1 (N = 3) (F5): Üç farklı alt fonksiyonun birleşiminden oluşmaktadır ve orta düzeyde karmaşıklık sunar. Optimal Değer (Fi): 1700
- Hibrit Fonksiyon 2 (N = 4) (F6): Dört farklı alt fonksiyon içeren bu yapı, daha karmaşık bir optimizasyon problemidir. Optimal Değer (Fi): 1600
- Hibrit Fonksiyon 3 (N = 5) (F7): Beş alt fonksiyondan oluşan bu fonksiyon, yüksek karmaşıklık içermektedir. Optimal Değer (Fi): 2100

3.4. Kompozisyon fonksiyonları

Kompozisyon fonksiyonları, farklı fonksiyonların ağırlıklı bir birleşiminden oluşmaktadır ve yüksek boyutlu optimizasyon problemlerinde algoritmaların performansını test etmek için kullanılır.

- Kompozisyon Fonksiyon 1 ($N = 3$) (F8): Üç alt fonksiyonun birleşiminden oluşur ve çok boyutlu keşif gerektirmektedir. Optimal Değer (Fi): 2200
- Kompozisyon Fonksiyon 2 ($N = 4$) (F9): Dört alt fonksiyonu birleştirerek daha zorlu bir yapı sunmaktadır. Optimal Değer (Fi): 2400
- Kompozisyon Fonksiyon 3 ($N = 5$) (10): Beş alt fonksiyonun birleşimiyle, en karmaşık optimizasyon problemlerinden birini temsil etmektedir. Optimal Değer (Fi): 2500

CEC 2020 test fonksiyonları, optimizasyon algoritmalarını çeşitli zorluk seviyelerinde ve farklı problem türlerinde test etmek için etkili bir araçtır. Bu fonksiyonlar, hem teorik hem de pratik optimizasyon çalışmalarında önemli bir referans noktası oluşturmaktadır. Bu çalışma, Cüce Firavun Faresi Optimizasyon (DMO) ve Nutcracker Optimizasyon Algoritması (NOA) performanslarının karşılaştırılmasında, bu test fonksiyonlarından faydalanmayı amaçlamaktadır.

CEC 2020 test fonksiyonlarının özetine ilişkin bilgiler Tablo 3'te sunulmaktadır.

Tablo 3. CEC 2020 ölçüt setindeki fonksiyonlar.

Fonksiyon No	Fonksiyonlar	Fi (Optimal Global Değer)
F1	Unimodal fonksiyon Shifted and Rotated Bent Cigar fonksiyonu	100
F2	Multimodal fonksiyon Shifted and Rotated Schwefel's Fonksiyonu	1100
F3	Shifted and Rotated Lunacek bi-Rastrigin fonksiyonu	700
F4	Expanded Rosenbrock's plus Griewangk's fonksiyonu	1900
F5	Hibrit fonksiyonlar Hibrit fonksiyon 1 (N = 3)	1700
F6	Hibrit fonksiyon 2 (N = 4)	1600
F7	Hibrit fonksiyon 3 (N = 5)	2100
F8	Composition fonksiyonlar Composition fonksiyon 1 (N = 3)	2200
F9	Composition fonksiyon 2 (N = 4)	2400
F10	Composition fonksiyon 3 (N = 5)	2500

BULGULAR VE SONUÇ

NOA ve DMO algoritmalarının CEC 2020 test fonksiyonları (F1-F10) üzerindeki karşılaştırmalı performans sonuçları tablo 4'te sunulmaktadır.

Tablo 4. Algoritmaların CEC 2020 karşılaştırma sonuçları.

No		Algoritmalar	
		NOA	DMO
F1	Ortalama	2.0290	6.2123
	Std. Sapma	2.7619	8.9624
F2	Ortalama	44.9742	4.2764
	Std. Sapma	24.2988	2.5000
F3	Ortalama	600.0001	2.1896
	Std. Sapma	4.3985	2.1223
F4	Ortalama	828.6505	321.2183
	Std. Sapma	6.7927	339.1482
F5	Ortalama	900.4002	228.3476
	Std. Sapma	1.2794	220.8289
F6	Ortalama	1812.2444	293.4755

	Std. Sapma	11.8444	246.947 9
F7	Ortalama	2020.223 0	211.083 3
	Std. Sapma	7.8268	161.782 2
F8	Ortalama	2220.615 2	289.526 0
	Std. Sapma	3.0764	218.975 8
F9	Ortalama	2480.781 2	257.583 8
	Std. Sapma	3.2792	243.443 4
F10	Ortalama	2500.421 0	303.237 6
	Std. Sapma	0.0659	256.498 7

Tablo 4'te görüldüğü üzere her bir fonksiyon için ortalama ve standart sapma değerleri sunulmuştur. Bu metrikler, algoritmaların doğruluğunu, kararlılığını ve optimizasyon süreçlerindeki etkinliğini değerlendirmek açısından önemlidir KAYNAKÇA. Genel olarak, NOA algoritması bazı test fonksiyonlarında DMO'ya göre daha iyi sonuçlar verirken, bazı fonksiyonlarda DMO'nun daha iyi performans gösterdiği görülmektedir. Bu durum, her iki algoritmanın da farklı problem türlerine göre güçlü ve zayıf yönlerinin bulunduğunu ortaya koymaktadır.

F1, F2 ve F3 fonksiyonlarının sonuçları incelendiğinde, her iki algoritma arasında çarpıcı farklılıklar göze çarpmaktadır. F1

fonksiyonunda, NOA'nın ortalama sonucu (2.0290) DMO'ya (6.2123) göre oldukça düşüktür. Standart sapma değerleri de benzer şekilde NOA'nın (2.7619) DMO'ya (8.9624) göre daha kararlı olduğunu göstermektedir. Bu durum, NOA'nın F1 üzerinde daha iyi keşif ve sömürü dengesine sahip olduğunu düşündürmektedir. Ancak, F2 ve F3 fonksiyonlarında durum tersine dönmektedir. F2'de DMO (4.2764) NOA'ya (44.9742) göre çok daha iyi bir ortalama performans sergilemiştir. Benzer şekilde, F3 fonksiyonunda da DMO (2.1896) daha iyi bir ortalama çözüme ulaşırken, NOA (600.0001) oldukça zayıf bir performans göstermiştir. Bu sonuçlar, NOA'nın F2 ve F3 fonksiyonları üzerinde yerel optima sıkışmasına daha fazla maruz kaldığını gösterebilir.

F4, F5 ve F6 fonksiyonlarına ilişkin sonuçlar, her iki algoritmanın daha karmaşık problem türleriyle nasıl başa çıktığını göstermektedir. F4 fonksiyonunda, DMO (321.2183) ortalama çözümü, NOA'nın (828.6505) çözümünden çok daha iyidir. Standart sapma değerleri de benzer bir eğilimi yansıtarak, DMO'nun bu fonksiyonda daha kararlı olduğunu ortaya koymaktadır. F5 fonksiyonunda ise DMO (228.3476) yine daha iyi performans gösterirken, NOA (900.4002) oldukça zayıf bir ortalama değer elde etmiştir. Bu, DMO'nun arama alanında daha etkili bir şekilde keşif yapabildiğini göstermektedir. F6 fonksiyonunda da benzer bir durum gözlemlenmektedir. DMO (293.4755) ortalama performansı, NOA'nın (1812.2444) değerine göre çok daha iyidir. Bu üç fonksiyon genelinde DMO, NOA'ya kıyasla daha başarılı sonuçlar elde etmiştir. Bu, DMO'nun karmaşık ve çok modlu fonksiyonlarda daha iyi bir keşif ve sömürü stratejisi izlediğini göstermektedir.

F7, F8, F9 ve F10 fonksiyonlarının sonuçları, daha zorlu ve yüksek boyutlu optimizasyon problemleriyle başa çıkma yeteneğini değerlendirmektedir. Bu fonksiyonlar üzerinde de DMO'nun NOA'ya göre belirgin bir üstünlüğe sahip olduğu görülmektedir. F7 fonksiyonunda, DMO (211.0833) daha iyi bir ortalama değer elde ederken, NOA (2020.2230) ile çok daha zayıf kalmıştır. Standart sapma değerleri, DMO'nun (161.7822) daha kararlı sonuçlar ürettiğini göstermektedir. F8, F9 ve F10 fonksiyonlarında da DMO, NOA'ya göre daha düşük ortalama ve standart sapma değerleriyle üstünlük sağlamıştır. Örneğin, F10'da DMO (303.2376) ortalama çözümü elde ederken, NOA (2500.4210) ile oldukça geride kalmıştır. Bu fonksiyonlar genelinde DMO'nun, yüksek boyutlu ve çok modlu problemlerde NOA'ya göre daha kararlı ve etkili çözümler ürettiği sonucuna varılabilir.

Genel olarak, NOA ve DMO algoritmalarının performanslarının CEC 2020 fonksiyon türüne göre değişiklik gösterdiği net bir şekilde görülmektedir. NOA, F1 fonksiyonunda daha iyi bir keşif ve sömürü dengesi kurarken, F2'den F10'a kadar olan fonksiyonlarda DMO'nun belirgin bir üstünlüğü vardır. Özellikle, DMO'nun daha düşük ortalama ve standart sapma değerlerine sahip olması, bu algoritmanın çözüm arayışında daha dengeli ve kararlı bir performans sergilediğini ortaya koymaktadır. Bu sonuçlar, DMO'nun yüksek boyutlu ve çok modlu problemler üzerinde daha etkin bir optimizasyon gerçekleştirdiğini göstermektedir. Buna karşın, NOA'nın F1'deki başarısı, bu algoritmanın bazı belirli problem türlerinde etkili olabileceğini, ancak daha karmaşık ve çok modlu problemlerde yerel optima sıkışmasına maruz kalabileceğini göstermektedir. Bu nedenle, her

iki algoritmanın da güçlü ve zayıf yönleri göz önünde bulundurularak, belirli bir problem türüne uygun algoritma seçimi yapılmalıdır.

REFERANSLAR

Abdel-Basset, M., Mohamed, R., Jameel, M., & Abouhawwash, M. (2023). Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. *Knowledge-Based Systems*, 262, 110248. <https://doi.org/10.1016/j.knosys.2022.110248>

Agushaka, J. O., Ezugwu, A. E., & Abualigah, L. (2022). Dwarf Mongoose Optimization Algorithm. *Computer Methods in Applied Mechanics and Engineering*, 391, 114570. <https://doi.org/10.1016/j.cma.2022.114570>

Alatas, B. (2019). Sports inspired computational intelligence algorithms for global optimization. *Artificial Intelligence Review*, 52(3), 1579-1627. <https://doi.org/10.1007/s10462-017-9587-x>

Altay, O. (2022). Chaotic slime mould optimization algorithm for global optimization. *Artificial Intelligence Review*, 55(5), 3979-4040. <https://doi.org/10.1007/s10462-021-10100-5>

Arslan, G., Aydemir, F., & Arslan, S. (2024). Enhanced license plate recognition using deep learning and block-based approach. *Journal of Scientific Reports-A*, 058, 57-82. <https://doi.org/10.59313/jsr-a.1505302>

Aydemir, S. B. (2022). Küresel Optimizasyon için Gauss Kaotik Haritası ile Kartal Optimizasyonu. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 34(1), 85-104. <https://doi.org/10.35234/fumbd.969335>

Bai, J., Li, Y., Zheng, M., Khatir, S., Benaissa, B., Abualigah, L., & Abdel Wahab, M. (2023). A Sinh Cosh optimizer. *Knowledge-*

Based Systems, 282, 111081.
<https://doi.org/10.1016/j.knosys.2023.111081>

Balochian, S., & Baloochian, H. (2019). Social mimic optimization algorithm and engineering applications. *Expert Systems with Applications*, 134, 178-191.
<https://doi.org/10.1016/j.eswa.2019.05.035>

Dörterler, S., Dumlu, H., Özdemir, D., & Temurtaş, H. (2024). Hybridization of Meta-heuristic Algorithms with K-Means for Clustering Analysis: Case of Medical Datasets. *Gazi Journal of Engineering Sciences*, 10(1), 1-11.
<https://doi.org/10.30855/gmbd.0705N01>

Dumlu, H., & Yavuz, G. (t.y.). *CEC 2019 Problemleri İçin Kendinden Uyarlamalı Bir Diferansiyel Gelişim Algoritması A Self-Adaptive Differential Evolution Algorithm for CEC 2019 Problems*.

Dursun, M. A., & Serttaş, S. (2024). A Multi-Metric Model for analyzing and comparing extractive text summarization approaches and algorithms on scientific papers. *DÜMF Mühendislik Dergisi*.
<https://doi.org/10.24012/dumf.1376978>

Eroz, E., & Tanyildizi, E. (2018). Güncel Metasezgisel Optimizasyon Algoritmalarının Performans Karşılaştırılması. *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, 1-16.
<https://doi.org/10.1109/IDAP.2018.8620764>

Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine Predators Algorithm: A nature-inspired

metaheuristic. *Expert Systems with Applications*, 152, 113377. <https://doi.org/10.1016/j.eswa.2020.113377>

Gunawan, D. L., & Kristian, Y. (2023). A Novel Approach to Flexible Multi-Resolution Image Compression using Deep Learning Based Autoencoders on Overlapping Image Patch. *Procedia Computer Science*, 227, 346-354. <https://doi.org/10.1016/j.procs.2023.10.533>

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.combiomed.2023.107389>

Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems*, 111, 300-323. <https://doi.org/10.1016/j.future.2020.03.055>

Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95, 51-67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>

Özen Kavas, P., Recep Bozkurt, M., Kocayiğit, İ., & Bilgin, C. (2023). Machine learning-based medical decision support system for diagnosing HFpEF and HFrEF using PPG. *Biomedical Signal Processing and Control*, 79, 104164. <https://doi.org/10.1016/j.bspc.2022.104164>

Sadollah, A., Eskandar, H., Bahreininejad, A., & Kim, J. H. (2015). Water cycle algorithm for solving multi-objective

optimization problems. *Soft Computing*, 19(9), 2587-2603.
<https://doi.org/10.1007/s00500-014-1424-4>

Yagmur, N., Dag, I., & Temurtas, H. (2024). A new computer-aided diagnostic method for classifying anaemia disease: Hybrid use of Tree Bagger and metaheuristics. *Expert Systems*, 41(8), e13528.
<https://doi.org/10.1111/exsy.13528>

Yavuz, G. (2022a). Diversified Position Update Equation-Based SSA with Refreshing-Gap Strategy for global optimization. *Journal of Computational Science*, 60, 101597.
<https://doi.org/10.1016/j.jocs.2022.101597>

Yavuz, G. (2022b). Senior Learning JAYA With Powell's Method and Incremental Population Strategy. *IEEE Access*, 10, 103765-103780. <https://doi.org/10.1109/ACCESS.2022.3210122>

Yavuz, G., Moghanjoughi, M. K., Dumlu, H., & Cakir, H. I. (2024). A Feature Selection Method Combining Filter and Wrapper Approaches for Medical Dataset Classification. *Vietnam Journal of Computer Science*, S2196888824500246.
<https://doi.org/10.1142/S2196888824500246>

BÖLÜM VIII

Mobil Uygulama İle Domates Yaprak Görüntülerinden Hastalık Teşhisi

Oğuzhan SEÇGEL¹
Kamil ORMAN²

Giriş

Dünya genelinde en çok tüketilen ve ekonomik olarak büyük öneme sahip tarım ürünlerinden birisi de domatestir. Günümüzde bu ürünün üretimi, çeşitli hastalıklar nedeniyle ciddi riskler altındadır. Bu hastalıkların erken teşhisi, hem ürün kaybını önlemek hem de tarım verimliliğini artırmak açısından büyük önem taşımaktadır. Geleneksel hastalık teşhis yöntemleri, genellikle uzman bilgisi gerektirir ve zaman alıcıdır. Ayrıca, bu yöntemlerin uygulanması için gerekli ekipman ve uzmanlık düzeyi domates üretimi yapılan her bölgede kolaylıkla bulunamayabilir. Bu bağlamda, teknolojinin

¹ Lisans Öğrencisi, Erzincan Binali Yıldırım Üniversitesi, Bilgisayar Mühendisliği Bölümü, Orchid: 0009-0002-3002-1059, oguzhan.secgel.19@gmail.com

² Doç. Dr., Erzincan Binali Yıldırım Üniversitesi, Bilgisayar Mühendisliği Bölümü, Orchid: 0000-0002-7236-9988, korman@erzincan.edu.tr

tarımda kullanımını, özellikle mobil teknolojilerin entegrasyonu, büyük bir potansiyel sunmaktadır. Görüntü işleme ve yapay zeka teknolojilerinin mobil platformlara entegrasyonu sayesinde, kullanıcılar artık tarlada anında hastalık teşhisi yapabilir. Bu çalışma, domates yaprakları üzerinde görülen hastalık belirtilerini tespit edebilen bir mobil uygulama geliştirmeyi amaçlamaktadır. Uygulama, kullanıcılara gerçek zamanlı veri sağlayarak, hastalıkların hızlı bir şekilde tespit edilmesine ve gerekli müdahalelerin zamanında yapılmasına olanak tanımaktadır.

Dünyada tarımsal üretimde oluşan kayıpların nedeni incelendiği zaman %9,1'inin hastalıklardan, %11,2'sinin zararlılardan, %14,7'sinin ise yabancı otlar kaynaklı olduğu görülmektedir. Bu oran tüm dünyadaki tarımsal üretimin üçte birine yakın bir orandır. Bu kayıplara %6-12 arasında değişebilen hasat sonrası kayıplar eklendiği zaman toplam kayıp oranı %50'lere yaklaşmaktadır (Kılıçarslan ve Pacal, 2023). Uygulama süreçlerine yönelik hazırlanan farklı algoritmalar ve yazılımların kullanımı ile tarımsal üretim süreçlerinde; bitkisel üretimin planlanmasında, bitki sınıflandırılmada, üretim veriminin tahmininde, bitkisel hastalıkların, zararlıların ve yabancı otların tespitinde gibi konu başlıklarında araştırmacılar tarafından çok sayıda çalışmalar ve araştırmalar yapılmaya başlanmıştır (Demir, Kula ve Uğurlu, 2021). Bu çalışmada lisans bitirme tezi olarak sunulan ve domates yapraklarından alınan görüntüye dayalı hastalık teşhisi yapabilen bir mobil uygulamanın geliştirilme süreci, kullanılan görüntü işleme algoritmaları ve yapay zeka modelleri, ayrıca uygulamanın test süreçleri ve başlangıç aşamasındaki performans sonuçları kapsamlı bir şekilde ele alınacaktır. Uygulamanın potansiyel etkileri, domates üretimi üzerindeki olası pozitif yansımaları ile birlikte tartışılacak ve gelecekteki araştırmalar için önerilerde bulunulacaktır.

Tarımda, görüntü işleme ve bilgisayarlı görme uygulamalarının, ekipman maliyetlerinin düşmesi ile hesaplama yeteneklerinin artması ve nitelikli gıda değerlendirme yöntemlerine olan ilginin artması ile yapay zekâ kullanımına yönelik ilgi giderek artmıştır (Akar ve ark., 2022, Mahajan ve Sardana, 2015). Yapay

zeka algoritmaları özellikle görüntü işleme uygulamalarında yaygın olarak kullanılmaktadır (Akar ve Aydın, 2017, Genç ve Akar, 2024, Aydın, 2023, Taş, 2021, Taş ve ark., 2021, Arslantaş ve Ökdem, 2024).

Domates Yaprak Hastalıkları

Domates yapraklarından görsel olarak tespit edilebilen hastalıkların belirti ve hastalık tanımlamasına ait bilgiler Tablo.1'd verilmiştir. Domates yapraklarındaki hastalıkları tanımlamak için EfficientNetV2 ve YOLOv5 modellerini entegre ederek yapılan çalışmada (Zhao ve ark., 2022), EfficientNetV2 modeli kullanılarak hastalıklı yaprakların sınıflandırılması gerçekleştirilmiş ve YOLOv5 algoritması ile hedef tespiti yapılmıştır. YOLOv5'in, domates yapraklarındaki hastalıkları yüksek doğruluk oranlarıyla tespit etmesi, tarımda hastalık yönetiminde büyük bir adım olarak değerlendirilmiştir.

Tablo 1. Domates Görsel Hastalık Belirtileri ve Nedenleri (Mokhtar ve ark., 2015)

Görsel Belirti	Hastalık veya Zararlı Türü
Siyah, kahverengi, koyu veya sulu noktalar	Bakteriyel benek-Bakteriyel nokta- Erken yanıklığı-Gri yaprak lekesi-Geç yanıklık-Septoria yaprak lekesi-Örümcek akarları-Domates benekli solgunluk virüsü
Koyu çizgiler	Tütün mozaik virüsü
Yeşillikteki delikler	Yaprak bitleri-Salyangozlar-Tütün boynuz kurdu-Domates kurdu-Domates boynuz kurdu
Düzensiz noktalar	Gri küf-Geç yanıklık
Yaprak kahverengileşmesi	Bakteriyel pamukçuk
Yaprak beneklenmesi	Salatalık mozaik virüsü-Tütün mozaik virüsü- Herbisit yaranlanması
Yaprak rulo	Yaprak bitleri-Kıvrıcık üst virüs-Herbisit
Morlaşan Damarlar	Yaralanma-Fizyolojik yaprak rulo- Patates yaprağı rulo
Spiral tasarımlar	Fosfor eksikliği
Yapışkan çiy (tatlı özsu)	Yaprak madencileri-Beyaz sinekler
Soyulmuş yapraklar, yaprak dökümü	Yaprak bitleri-Beyaz sinekler
Beyaz noktalar	Tütün boynuz kurdu-Domates kurdu-Domates boynuz kurdu
Sararma ve solma	Yaprak küfü-Magnezyum eksikliği-Toz halinde küf-Örümcek akarları

Şekil 1 ve Şekil 2'de ise sırasıyla sağlıklı ve hastalıklı domates yaprak görüntüleri verilmiştir.



Şekil 1. Hastalıklı Domates Yaprakları



Şekil 2. Sağlıklı Domates Yaprakları

Geliştirilecek olan mobil uygulama da teşhis edilmesi planlanan domates hastalıkları ise aşağıda sıralanmıştır.

1. Erken Yanıklık (Early Blight): Domates bitkilerinde yaygın olarak görülen bir fungal hastalıktır. Yapraklarda kahverengi lekeler ve kenarlarda sararmalarla karakterizedir.

2. Sağlıklı (Healthy): Bu sınıf, hastalıklardan etkilenmeyen sağlıklı domates yapraklarını temsil eder. Yapraklar tipik yeşil renkte ve herhangi bir leke veya belirti göstermez.

3. Geç Yanıklık (Late Blight): Domates bitkilerinde yaygın olarak görülen bir fungal hastalıktır. Yapraklarda koyu lekeler, siyah noktalar ve kahverengi çürümeler oluşur.

4. Yaprak Minerleri (Leaf Miner): Yaprak minerleri, yaprakların iç kısımlarında beslenen küçük böceklerdir. Yapraklarda kahverengi izler ve benekler oluşur.

5. Yaprak Küfü (Leaf Mold): Bu fungal hastalık, yaprakların üst yüzeyinde beyaz veya sarı renkli küf tabakaları oluşturur. Yapraklar üzerinde lekeler ve deformasyonlar görülebilir.

6. Mozaik Virüsü (Mosaic Virus): Mozaik virüsü, domates bitkilerinde yaygın olarak görülen bir viral hastalıktır. Yapraklarda renk değişimleri, mozaik desenleri ve deformasyonlar oluşur.

7. Septoria: Domates bitkilerinde yaygın olarak görülen bir fungal hastalıktır. Yapraklarda kahverengi lekeler, sararma ve yaprak dökülmeleri görülür.

8. Örümcek Akarı (Spider Mites): Yaprakların alt kısımlarında bulunan ve bitki özsuğunu emen küçük örümcek benzeri böceklerdir. Yapraklarda soluk lekeler ve sararmalar oluşur.

9. Sarı Yaprak Kıvrılma Virüsü (Yellow Leaf Curl Virus): Sarı yaprak kıvrılma virüsü, domates bitkilerinde yaygın olarak görülen bir viral hastalıktır. Yapraklarda sararma, vırlma ve kıvrık büyüme belirtileri görülür.

Kullanılan Teknolojiler

Bu çalışma kapsamında, domates yaprağı hastalığı teşhisini gerçekleştirmek için, veri setlerini Kaggle platformundan elde edilmiş ve verileri etiketlemek için Roboflow platformu kullanılmıştır.

Kullanılan görüntü işleme modelinin eğitimi sürecinde ise YOLOv8 algoritması kullanılmıştır. Bu model, domates yaprağı hastalıklarını tespit etmek için kullanılan etkili bir nesne algılama algoritmasıdır. Eğitim süreci Google Colab üzerinde gerçekleştirilmiştir. Google Colab, bulut tabanlı bir Jupyter defteri hizmeti sunarak model eğitimi hızlandırmış ve hesaplama kaynaklarını sağlamıştır.

Kullanıcı arayüzü aşamasında ise mobil uygulamayı geliştirmek için Flutter framework'ü kullanılmıştır. Flutter, hızlı ve etkili bir şekilde platformlar arası mobil uygulamalar geliştirmesine olanak tanıyan açık kaynaklı bir SDK'dır. Bu sayede, domates yaprağı hastalığı teşhisi için geliştirilen mobil uygulama hem Android hem de iOS platformlarında yayınlayabilecek ve geniş bir kullanıcı kitlesine ulaşabilecektir.

Bu teknolojilerin mobil platformlara entegrasyonu, hastalık teşhis sürecini otomatikleştirme ve tarım verimliliğini artırma açısından kullanıcılara yardımcı olabilecek potansiyele sahip olacaktır. Yapılan bu çalışma, tarımsal uygulamalarda teknoloji kullanımının potansiyelini ve etkisini göstermeyi hedeflemektedir.

Veri Seti ve Ön İşleme

Domates yapraklarındaki hastalıkların tespiti için toplamda 4212 görsel kullanılmıştır (Tomato Leaves Veri Seti). Tomato Leaf Disease veri setinin %2'lik kısmı test için rastgele seçilerek ayrılmış ve test veri seti olarak adlandırılmıştır. Test veri setinin içindeki her görüntü için model çalıştırılmış ve tahmin sonuçları kaydedilmiştir. Modelin performansı, gerçek etiketler ile modelin tahmin ettiği etiketler arasında karşılaştırmalar yapılarak değerlendirilmektedir.

Değerlendirme sürecinde, modelin performansını ölçmek için mAP (Mean Average Precision), Hassasiyet (Precision), Duyarlılık (Recall) ve F1-Skoru gibi metrikler kullanılmıştır.

mAP (mean Average Precision), nesne tespiti ve bilgi çıkarma gibi makine öğrenimi uygulamalarında model performansını değerlendirmek için yaygın olarak kullanılan bir metriktir. mAP, her sınıf için hesaplanan Average Precision (AP) değerlerinin ortalamasını olarak hesaplanır.

$$AP = \sum_n (R_n - R_{n-1}) P_n$$

Burada P_n , n. precision değerini, R_n , n. recall değerini ve R_{n-1} ise (n-1). recall değerini ifade etmektedir.

$$mAP = \frac{1}{N_n} \sum_{i=1}^N AP_i$$

Burada N , Toplam sınıf sayısını ve AP_i ise i. sınıf için Average Precision değerini ifade etmektedir.

Hassasiyet (Precision) metriği ise bir sınıflandırma modelinin pozitif tahminlerinin ne kadarının doğru olduğunu ölçer. Precision, doğru pozitif (true positive, TP) tespitlerin, tüm pozitif tespitlere oranıdır.

$$Precision = \frac{Doğru Pozitif (TP)}{Doğru Pozitif (TP) + Yanlış Pozitif (FP)}$$

Burada Doğru Pozitif (TP): Modelin doğru bir şekilde pozitif olarak sınıflandırdığı örnekleri, Yanlış Pozitif (FP) ise Modelin pozitif olarak sınıflandırdığı, ancak aslında negatif olan örnekleri ifade etmektedir.

Duyarlılık (Recall), modelin gerçekten pozitif olan tüm örnekleri ne kadar iyi tespit ettiğini gösterir. Başka bir deyişle, modelin tüm pozitif örnekler arasından kaç tanesini doğru bir şekilde pozitif olarak sınıflandırdığını ölçer. Bu metrik, doğru pozitif (true

positive) oranını değerlendirir ve yanlış negatifleri (false negatives) minimize etmeyi hedefler.

$$Recall = \frac{Doğru\ Pozitif\ (TP)}{Doğru\ Pozitif\ (TP) + Yanlış\ Negatif\ (FN)}$$

Bu metrikler, modelin hastalık tahminindeki başarısını ve doğruluğunu değerlendirmek amacıyla hesaplanmaktadır.

Veri setinde kullanılan görsellerin dağılımı ise şu şekildedir:

- Erken Yanıklık (Early Blight): 1.839 adet
- Mozaik Virüsü (Mosaic Virus): 1.750 adet
- Septoria: 1.632 adet
- Yaprak Küfü (Leaf Mold): 1.625 adet
- Sarı Yaprak Kıvrılma (Yellow Leaf Curl): 1.611 adet
- Yaprak Minerleri (Leaf Miner): 1.357 adet
- Örümcek Akarı (Spider Mites): 1.229 adet
- Geç Yanıklık (Late Blight): 1.193 adet
- Sağlıklı (Healthy): 730 adet

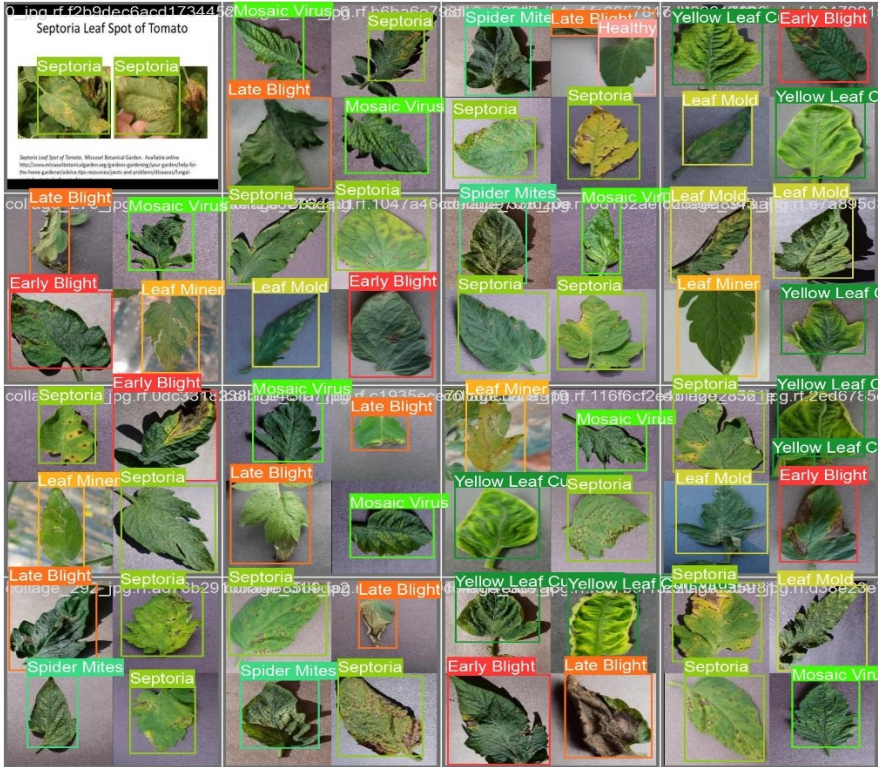
Ön işleme adımında, ilk olarak veri setinin temizleme işlemleri yapılmıştır. Domates yapraklarının görüntülerinden oluşan veri setinde bulunan gereksiz sütunlar ve etiketler atılmıştır. Bu görseller Roboflow platformu üzerinden aşağıdaki işlem adımları kullanılarak işlenmiştir.

- Otomatik yönlendirme uygulanmıştır.
- Yeniden boyutlandırma yapılarak, boyutlar 640x640 piksel olacak şekilde sabitlenmiştir
- Veri artırma işlemi gerçekleştirilmiştir.

Veri artırma işlemi şu şekilde yapılmıştır:

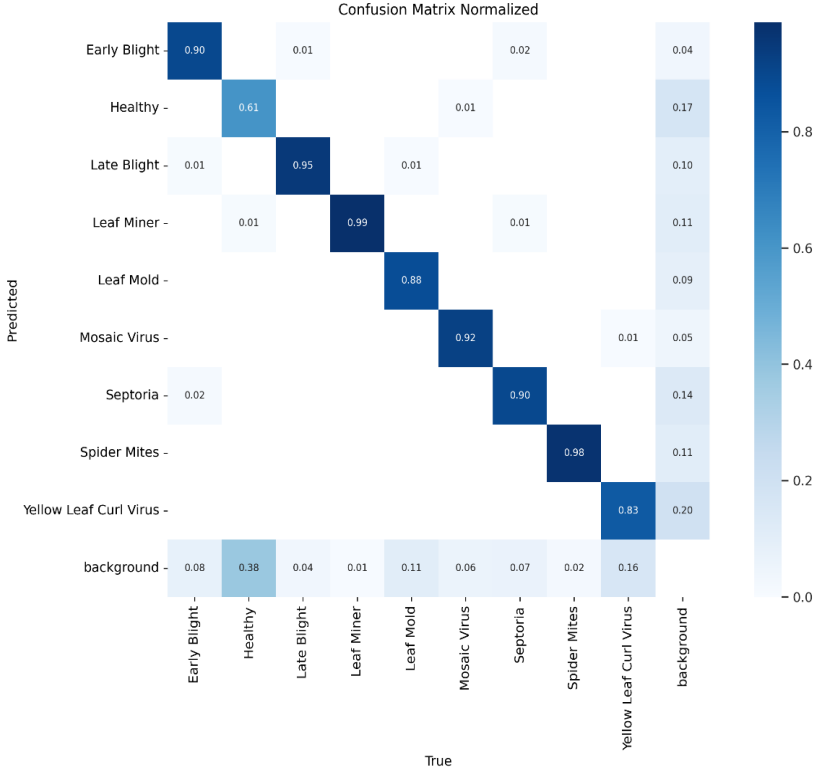
- Görüntü kesme oranı minimumda %0, maksimumda %2 olacak şekilde belirlenmiştir.
- Parlaklık değişimi -20% ile +20% arasında olacak şekilde belirlenmiştir.
- Bulanıklık efekti maksimum 1 piksel olacak şekilde uygulanmıştır.

Toplamda 3 kat veri artırma (augmentation) yapılarak eğitim setindeki görsel sayısı 10,746'ya ulaştırılmıştır. Bu görsellerin 9,801'i eğitim setinde, 783'ü doğrulama setinde ve 162'si test setinde kullanılmıştır.



Şekil 3. Test Setinin Hastalık Sınıflandırma Sonuçları

Roboflow üzerinden alınan API linki ve API anahtarı (API key) ile istek yoluyla model test edilmiştir. Şekil 3 ve Şekil 4'te kullanılan veri setinin test ve test sonucuna ilişkin görseller sunulmuştur.



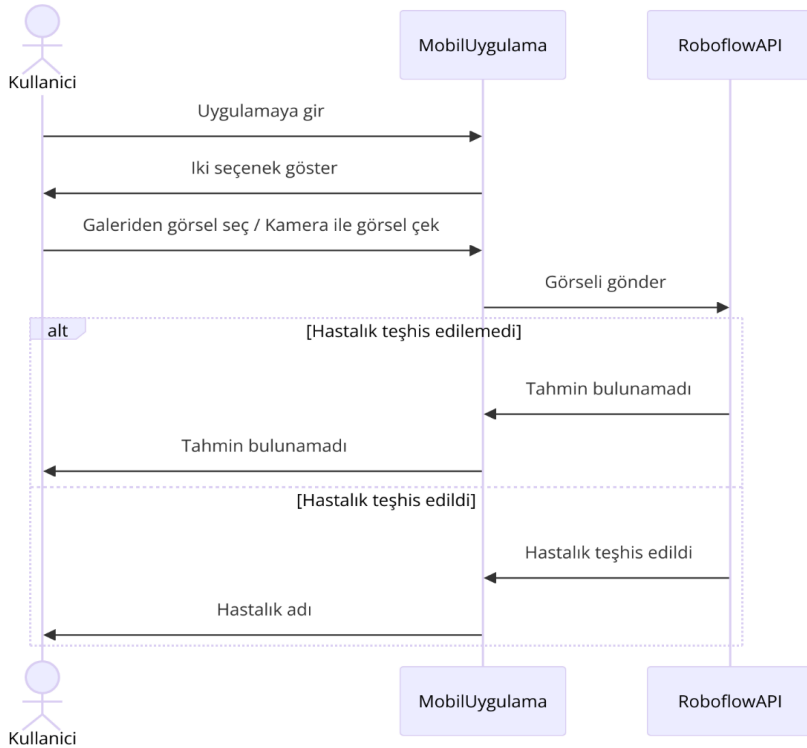
Şekil 4. Confusion Matrix Sonuçları

Mobil Uygulama ve Entegrasyonu

Mobil uygulamanın geliştirilmesi ve entegrasyonu aşamasında, geliştirilen modelin kullanıcılara sunulması için bir API oluşturulmuştur. Mobil uygulama, kullanıcının telefonundan çekilen veya galeriden seçilen domates yaprağı görüntüsünü API'ye

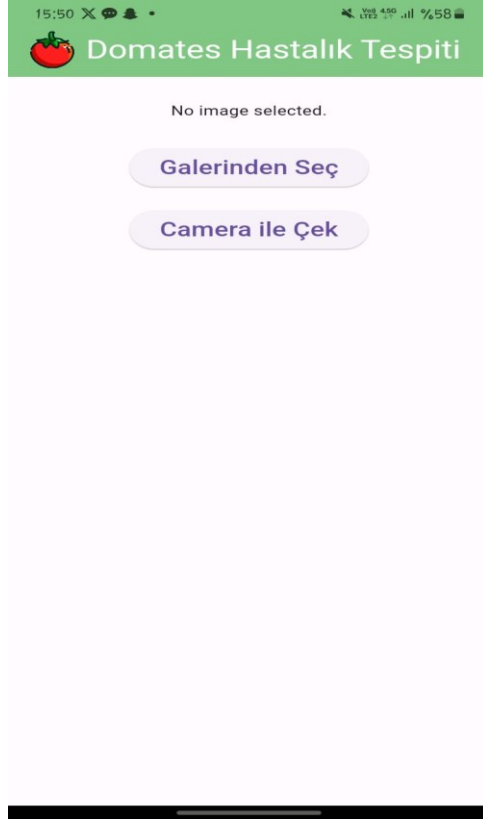
gönderir. API, bu görüntüyü işleyerek hastalık teşhisini gerçekleştirir ve sonuçları mobil uygulamaya geri gönderir. API entegrasyonu, mobil uygulamanın Flutter ile geliştirilmesi sayesinde hem Android hem de iOS platformlarında sorunsuz çalışmasını sağlar. Bu süreç, uygulamanın kullanıcı dostu olmasını ve geniş bir kullanıcı kitlesine hitap etmesini sağlamaktadır. Şekil 5'te geliştirilen mobil uygulamaya ait sistemin UML diyagramı verilmiştir.

Geliştirilen mobil uygulamanın ana ekran görüntüsüne ait görseller şekil 6'da verilmiştir.



Şekil 5. Sistem UML Tasarım Diyagramı

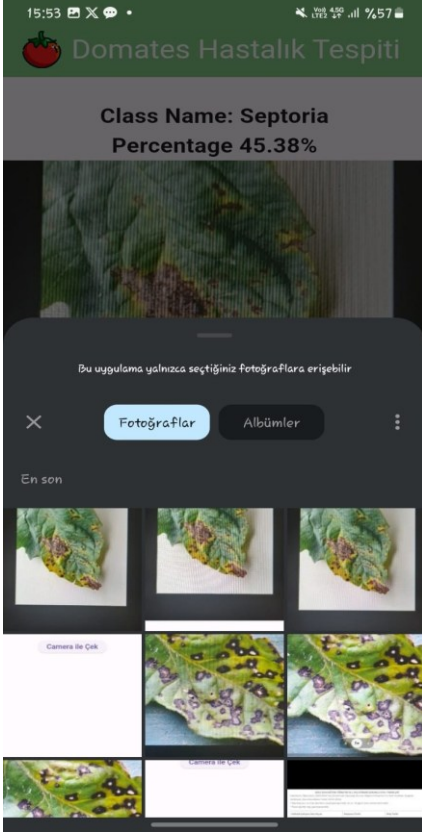
Uygulama ana ekranında kullanıcıların hastalık teşhisinde kullanılmak üzere domates yaprak görüntüsünü mobil cihazın galerisinden ya da mobil cihazın kamerasından yükleyebilmeleri için ilk seçenek ekranı bulunmaktadır (Şekil 7).



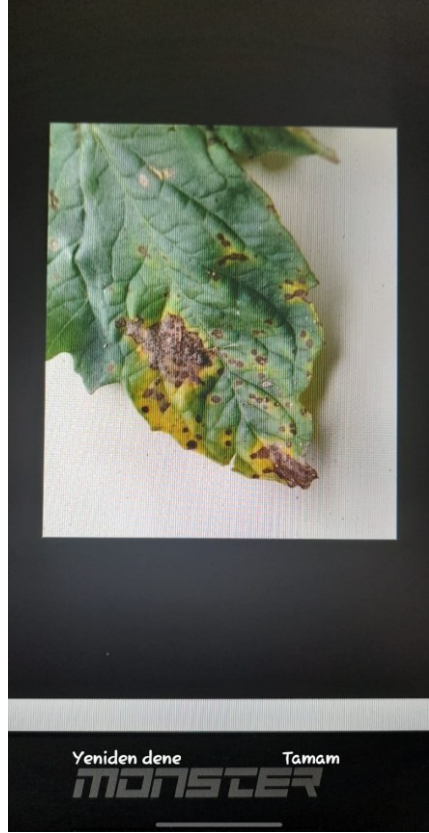
Şekil 6. Mobil Uygulamanın Ana Ekran Görüntüsü

Bu sayede kullanıcılar mevcut hastalıklı yaprak görüntülerinden uygulamanın tahmin doğruluğunu test edebilmesi ya da kullanıcıya başka bir cihazdan gönderilen görüntüleri kullanarak veya mobil cihazın kamerasından alınacak görüntü ile

hastalık tespiti yapabilmelerinin sağlanması amaçlanmıştır. Şekil 8’de ise sonuç ekranı gösterilmektedir.

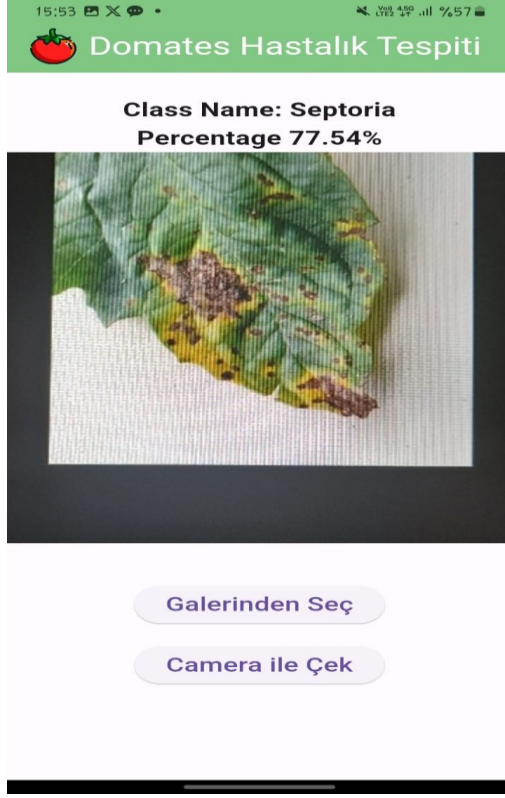


(a)



(b)

Şekil 7. Mobil Cihaz Galeriden Görsel Seçme (a) ve Kameradan Görsel Seçme (b)



Şekil 8. Sonuç Ekranı

Sonuç ve Tartışma

Bu çalışma, mobil uygulama aracılığıyla domates yaprağındaki hastalıkların teşhis edilmesi amacıyla geliştirilen bir yöntemin detaylarını sunmaktadır.

Geliştirilen mobil uygulama tarım sektöründeki profesyoneller ve amatör bahçıvanlar gibi geniş bir kullanıcı kitlesi için tasarlanmıştır. Kullanıcılar, domates yapraklarında görülen çeşitli hastalıkların hızlı ve etkili bir şekilde teşhis edilmesi için uygulamayı kullanabilirler. Bu sayede domates yapraklarında görülebilecek hastalıkların erken teşhis edilmesi ve hızlı bir şekilde

müdahale edilmesi sağlanarak, ürün verimliliği artırabilir ve ürün kayıpları azaltabilecektir.

Uygulama, görüntü işleme ve yapay zeka tekniklerini birleştirerek, domates yapraklarındaki ele alınan hastalıkları tanıma yeteneğine sahiptir. Yapay zeka ve görüntü işleme teknolojilerin mobil platform üzerinde bir araya gelmesi, uygulamanın hızlı, doğru ve güvenilir sonuçlar üretmesini sağlayacaktır.

Geliştirilen mobil uygulama aracılığıyla domates yaprağı hastalıklarının yerinde ve doğrudan kullanıcılar tarafından kendi mobil cihazları kullanılarak teşhisi konusunda bir adım atılmıştır. Ayrıca, gelecekte yapılması planlanan çalışmalar ile uygulamanın daha da geliştirilmesi mümkündür. Planlanan çalışmalardan biri de veri setinin genişletilmesidir bu sayede geliştirilen mobil uygulamanın daha fazla hastalığı tanımasını sağlayabilir. Diğer taraftan kullanıcı arayüzünün iyileştirilmesi ve mobil uygulamanın diğer bitki hastalıklarının teşhisi için uyarlanması gibi konular da üzerinde durulması gereken alanlardır.

KAYNAKÇA

Akar F., Aydın Y., Bülgen A. (2022). Resnet Modelleri İle Mısır Yapraklarında Hastalık Tespiti. International Informatics Congress (IIC2022), Batman, Türkiye, ss.381-391

Akar, F., & Aydın, Y. (2017). Comparison of Interest Point-Based Features in Object Recognition Applications. In 8th International Advanced Technologies Symposium (IATS'17), Elazığ, Türkiye (Vol. 19, No. 22, pp. 3553-3556).

Arslantas, H., & Okdem, S. (2024). Indoor Localization with an Autoencoder based Convolutional Neural Network. IEEE Access.

Aydin, Y. (2023). A Comparative Analysis of Skin Cancer Detection Applications Using Histogram-Based Local Descriptors. *Diagnostics*, 13(19), 3142.

Demir, Ü., Kula, N., & Uğurlu, B. (2021). Tarımda Yapay Zekâ Kullanımına Yönelik Karar Destek Modeli Önerisi: Domates Zararlısı Tespiti Örneği. *Lapseki Meslek Yüksekokulu Uygulamalı Araştırmalar Dergisi*, 2(4), 91-108.

Genç, M., & Akar, F. (2024). Detection of Lung Cancer Cells Using Deep Learning Methods. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 13(2), 445-459.

Kılıçarslan, S., & Pacal, I. (2023). Domates Yapraklarında Hastalık Tespiti İçin Transfer Öğrenme Metotlarının Kullanılması. *Mühendislik Bilimleri Ve Araştırmaları Dergisi*, 5(2), 215-222.

Mokhtar, U. et al. (2015). SVM-Based Detection of Tomato Leaves Diseases. In: Filev, D., et al. *Intelligent Systems'2014. Advances in Intelligent Systems and Computing*, vol 323. Springer, Cham.

Taş, M. B. H., (2021). Bulut Tabanlı Derin Öğrenme Kullanarak İnsansız Hava Araçları için Gerçek Zamanlı Nesne

Algılama, Yüksek Lisans Tezi, Atatürk Üniversitesi, Fen Bilimleri Enstitüsü, Erzurum.

Taş, M. B. H., Irmak, M. C., Turan, S., & Haşiloğlu, A. (2021, September). Real-time puddle detection using convolutional neural networks with unmanned aerial vehicles. In 2021 6th International Conference on Computer Science and Engineering (UBMK) (pp. 598-602). IEEE.

Zhao, S., Peng, Y., Liu, J., & Wu, S. (2021). Tomato leaf disease diagnosis based on improved convolution neural network by attention module. *Agriculture*, 11(7), 651.

<https://www.kaggle.com/datasets/ashishmotwani/tomato>

<https://app.roboflow.com>

<https://docs.roboflow.com/deploy/hosted-api/custom-models/object-detection>

<https://colab.google>

BÖLÜM IX

Spor Kulüpleri İçin Hizmet Olarak Beyaz Etiketli Platform

Ozan ÜST¹
Robbert OOMENS²

GİRİŞ

Spor, farklı kültür ve statüden milyonlarca insanı farklı yerlerde olsa bile bir noktada buluşturabilen, tüm dünyanın aynı anda tüketebildiği bir üründür [1]. Spor kulüpleri, taraftarları ile ne kadar çok iletişim kurup, onlar hakkında ne kadar çok bilgi sahibi olursa beklentilerini de o ölçüde belirleyip, o yönde hizmet sunarak taraftarın etkileşimine olumlu katkıda bulunurlar [2]. Sporun endüstrileşme süreci, geçtiğimiz yüzyılın sonlarına doğru futbol özelinde yaygınlaşıp popüler kültür haline dönüşmesiyle başlayıp, sonrasında diğer endüstri alanlarıyla rekabet eder seviyeye

¹ Senior Software Developer, Triodor R&D Center

² Product Director, Azerion

ulařmıřtır. Spor endüstrisi, gelişen teknoloji ve dijital platformların da etkisiyle milyarlarca kiřiye ulařarak, toplumsal ve ekonomik bir boyut kazanmıř, bu da bu sektörde gelirlerin artmasını saęlamıřtır.

Spor endüstrisinde amaç, tüketicilerin spor ile ilgili olan ürünleri ve hizmetleri tüketebilecekleri bir pazar alanı yaratmaktır [3]. Spor kulübü taraftarları müsabakalara katılarak, kulübün ürün ve hizmetlerini satın alarak veya internet sitelerine ve sosyal medya hesaplarına üye olarak takımlarıyla aktif bir etkileşim içerisine girebilirler. Profesyonel spor kulüpleri açısından, taraftarların bilet, ürün, hizmet satın alma ve spor etkinliklerine katılma kararlarını etkileyen faktörlerin tespit edilmesi önemlidir. Çünkü, taraftarlarının istek ve ihtiyaçlarını doğru şekilde analiz edip bu bilgilerle taraftar katılımını artırmaya yönelik etkili stratejiler oluşturarak gelirlerini ve marka değerini artırabilirler [4]. Bir firmanın müşteri tabanını oluşturmak ve bağlılığını sürdürmek, firmanın yalnızca “yeniden satın alma davranışının” ötesine bakmasını gerektirebilir. Taraftar katılımı/baęlılığı da, spor bağlamında müşteri katılımının özel bir biçimi olarak görülebilir. Bu baęlılık davranışlarını kolaylařtırmak, aynı şekilde bir spor ürününü yeni pazarlara sunarken de büyük önem taşımaktadır [5]. Her spor kulübünün ihtiyaçları ve hedefleri farklı olduğundan taraftar uygulamalarının, kulübün özelliklerine ve hedeflerine uygun şekilde özelleştirilebilir olması büyük bir avantaj sağlamaktadır. Türkiye’de birçok spor kulübünün, büyük kulüpler de dahil olmak üzere, ya mobil uygulaması bulunmamakta ya da mevcut uygulamaları eksik ve verimsiz çalışmaktadır. Bu durum, spor kulüplerinin dijital gelir fırsatlarını değerlendirememesine ve taraftarlarıyla dijital baę kuramamasına neden olmaktadır.

Çalışmamızın amacı, geliştirilecek platform ile spor kulüplerinin taraftarlarıyla daha yakın bir ilişki kurmaları, uygulama içi satın alımlarla gelirlerini arttırmaları ve eldeki geçmiş verinin kullanılmasıyla taraftar deneyimini geliştirmektir. Uygulama geliştirmek ve yönetmek zaman alıcı ve maliyetli olabileceği için bu proje çıktısının beyaz etiket çözümü, spor kulüplerinin strateji belirlemelerinde önemli bir rol oynayacaktır.

YÖNTEM

Çalışma kapsamında “gamification” (oyunlaştırma) özellikleri eklenmiş hibrit bir uygulama çerçevesi geliştirmek için “Unity Engine” kullanılmıştır. Buradaki amacımız, hibrit uygulama yapısı ve Unity motoru kullanan oyunlaştırma ile iki tarafın da en iyi özelliklerini birleştirerek, diğer oyun motorlarının limitlerini aşmaktır. API uç noktalarını bağlama ve yükleme, doku verilerini önbellekleme, canlı veri güncellemeleriyle ilgilenme ve kullanıcıya yetkilendirme, kimlik doğrulama ve diğer mesajları iletme özellikleri içeren bir çerçeve geliştirilmektedir. Bu çerçeve aynı zamanda, çevrimdışı veri depolayabilecek, gösterebilecek ve canlı veriyi analiz edip bağımlılıklarına karar vererek, uygulama ve oyunlaştırma fonksiyonallikleri arasında stabil ve hızlı geçiş sağlayacak bir yapı oluşturacaktır. Bir yandan, uygulamanın kullanıcı arayüzünde reklam göstermek için bir çözüm geliştirilmektedir; sponsor ve yerel reklamlar için ayrı hatlar kullanılacak ve reklamlar kendi arka yüzümüz veya Google AdManager aracılığıyla gösterilecektir. Böylelikle diğer uygulamalarda uygulanan basit bir üst üste gelme olayının önüne geçilmesi planlanmaktadır.

Unity, mobil için optimize edilmiş entegre oyun geliştirme araçları sunmakta ve endüstrideki popülerliği mevcut teknolojiden yararlanmamıza olanak tanımaktadır. Proje, hem oyunlaştırma hem de uygulama geliştirmeyi içerdiği için oyunlaştırma tarafındaki ihtiyaçlarımızı destekleyebilecek oyun motoru olarak Unity'nin kullanılması tercih edilmiştir. Böylelikle, kullanıcının oyunları oynamak için uygulamadan ayrılmasına gerek kalmadan uygulama içinde etkileşim oyunlarını dahil edebilmesi mümkün olacaktır. Burada geliştireceğimiz oyunlar 3D olacağı için yerel bir çerçeve ile aynı kaliteyi elde etmemiz mümkün olmayacaktı. Bu yüzden çerçeve çözümümüz; önbelleğe almayı optimize etmek ve uygulama ile oyunlaştırma katmanları arasında kesintisiz veri iletişimini sağlamak için Veri > Sistemler > Görevler > Görünümler tasarım modelini kullanmaktadır. Burada;

- Veri katmanı; uygulamanın durumudur,
- Sistemler katmanı; mantıksal işlemleri yönetir ve veriyi yönlendirir,
- Görevler katmanı; birleştirilmiş bir operasyon için bir dizi ayrılmış sistem çalıştırır,
- Görünümler katmanı ise; verileri kullanıcıya görselleştirir ve kullanıcının, sistemleri ve görevleri kullanarak veri ile etkileşime geçmesine izin verir.

Söz konusu platform için yapım hattını içeren sağlam bir temel oluşturulmuştur. Uygulama içerisine, esnekliği sayesinde birçok menüye de uygulanabilen “yerli reklamcılık” entegre edilmiştir. Oyunlaştırma katmanını genişletirken, uygulamamızı en iyi şekilde yüklemek için bir varlık yükleme çözümünün eklenmesi gerekmiştir. Ana uygulamayı askıya alıp onunla iletişim kurarken bir yandan

oyunun içinde performansı sürdürebilecek bir sistem gereklidir. Kullanıcı deneyimini zenginleştiren ve kullanıcı bağlılığını artıran birçok yeni özellik entegre edilmiştir. Uygulamanın oyunlaştırma katmanını genişletilerek genel fonksiyonalitesini arttırmak için çalışmalar gerçekleştirilmiştir. Kullanıcıların favori spor kulüpleriyle etkileşime girebileceği tüm bileşenler tek bir merkezde oluşturulmuştur. Uygulama; günlük görevler, kulüp puanları, rozet biriktirme, şans çarkı, oyunlar ve lider tablolarından oluşan bir oyunlaştırma katmanına sahiptir.

Yapılan bu geliştirmeler süresince canlı maçlara bağımlı olunması nedeniyle test aşamalarında birtakım teknik zorluklarla karşılaşmıştır. Kulüpler arasındaki veri uyumsuzlukları, eksik oyuncu varlıkları, yanlış puanlama ve grafik sorunları da çözülmesi gereken diğer zorluklar arasındadır. Kullanıcı erişilebilirliğini ve rahatlığını artırmak için farklı platformlarda (iOS ve Android) “Deep linking” (derin bağlantı) özelliği kullanılmıştır. Bu özellik, kullanıcıların anlık bildirimlerden doğrudan uygulama içindeki ilgili ekranlara sorunsuz bir şekilde gitmesine olanak tanıyarak genel kullanıcı deneyimini geliştirmektedir.

SONUÇLAR

Çalışmanın çıktısı, spor kulüplerine uçtan uca bütünsel bir taraftar bağlılık aracı ve gelir modeli sunmaktadır. Spor kulüplerine ekstra bir geliştirme yükü ve işgücü ihtiyacı oluşturmadan kişiselleştirilmiş taraftar veri tabanına erişim imkanı sağlamaktadır. Taraftarlar, uygulama üzerinden oyun oynamaktan bilet satın almaya, maç skoru tahmininden spor içeriklerini izlemeye kadar her türlü hizmete/içeriğe erişebilirler. Ayrıca, spor kulüplerinin ana gelir sağlayıcıları olan sponsorlar ve reklamverenler için dijital reklam

envanteri ve dijital reklam satış algoritması oluşturulmuş şekilde uygulama anahtar teslim olarak kulüplere sunulmakta; kulüplerin yazılımı ayrı servis sağlayıcı, içeriği ayrı içerik sağlayıcı, reklam teknolojilerini ayrı medya ajansı ile farklı kulvarlarda yönetmesine ihtiyaç kalmamaktadır. Dolayısıyla geliştirilen bu platform, bir spor kulübünün taraftar, sponsor, reklamveren ve içerik üreticisi bileşenlerinin hepsini kapsayarak bu unsurları bütünleşik bir şekilde adreslemektedir. Ayrıca proje çıktısı, bir beyaz etiket platform hizmeti şeklinde sunulmaktadır.

KAYNAKÇA

[1] Yavaş, Ö. (2005). Sporun ekonomi içindeki yeri ve spor pazarlama: Üç büyük spor kulübünde uygulamalı bir araştırma. Yüksek Lisans Tezi. Trakya Üniversitesi Sosyal Bilimler Enstitüsü.

[2] Goldwasser, R.D. (2007). How can the internet be successfully used by a professional sports team to manage and develop their fan relationships?: A case study of the New Orleans Hornets Doctoral dissertation. Grenoble Ecole de Management, USA.

[3] Smith, A.C.T. (2008). Introduction to Sport Marketing. Oxford, USA.

[4] Dittmore, S.W., Stoldt, G.C. and Greenwell, T.C. (2008). Use of an organizational weblog in relationship building: The case of a major league baseball team. International Journal of Sport Communication, 1(3), 384-397.

[5] van Doorn, J., Lemon, K.N., Mittal, V., Nass, S., Pick, D., Pirner, P., and Verhoef, P. C. (2010). Customer engagement behavior: Theoretical foundations and research directions. Journal of Service Research, 13, 253–266.

BÖLÜM X

Akıllı Şehirler: Bibliyometrik Analiz

Nevin AYDIN¹

Giriş

Akıllı şehirler, vatandaşların yaşam kalitesini ve kentsel hizmetlerin verimliliğini artırmak için teknoloji ve verilerden yararlanır. Nesnelerin interneti (IoT) ve makine öğrenimi, büyük ölçekli veri toplama, analiz ve karar almaya yönelik akıllı şehirlerde çözüm üretme teknolojileri olarak ortaya çıkmıştır. Bu teknolojiler daha yaşanabilir, sürdürülebilir ve verimli şehirlerin yaratılmasında büyük potansiyele sahiptir. Buna karşılık, akıllı şehirlerin potansiyelini gerçekleştirmek için ele alınması gereken veri gizliliği, güvenlik ve etik konular ile ilgili zorluklar devam etmektedir.

Akıllı şehirler, Nesnelerin İnterneti (IoT) sensörleri, bağlı cihazlar, yapay zeka (AI) ve gerçek zamanlı olarak çeşitli kaynaklardan veri toplamak ve analiz etmek için büyük veri analitiği gibi çeşitli teknolojilerden faydalanır. Akıllı bir şehrin amacı, bu

¹ ¹Prof. Dr., Artvin Çoruh Üniversitesi, İİBF, İşletme Bölümü, nevin.aydin@gmail.com

verileri ulaşım, enerji, su yönetimi, atık yönetimi ve kamu güvenliği gibi kentsel hizmetlerin verimliliğini ve etkinliğini artırmak için kullanılmaktadır. Şehirlerin geleceğinin teknolojik, sosyal ve çevresel eğilimlerin bir birleşimi olarak şekillendirilmesi gereklidir.

Akıllı şehir kavramları, kentleşmenin getirdiği zorluklara ve sürdürülebilir kentsel gelişime duyulan ihtiyaçtan ortaya çıkmıştır. Nesnelerin İnterneti, yapay zeka ve büyük veri analitiği gibi çeşitli teknolojilerin birlikte kullanımı, kentsel hizmetlerin verimliliğini artırmak, kaynak tüketimini azaltmak ve yaşam standardını yükseltmek gibi amaçları hedeflemektedir; bu vatandaşlar için daha iyi bir yaşam kalitesi yaratır. Teknolojilerin gelişimiyle, akıllı şehirler, akıllı tarım, akıllı endüstri, akıllı çiftçilik, akıllı sağlık hizmeti, akıllı trafik ve akıllı yayalar gibi çeşitli uygulamalardan veri toplayıp, yaşam standardını iyileştirmek için verileri analiz ederek entegre edebilirler (Wang vd., 2018).

Bibliyometrik Analiz

Bibliyometrik analiz, belirli bir konu üzerinde üretilen akademik yayınlar arasındaki ilişkilerin sayısal analizi olarak tanımlanır. Yayın veri tabanlarından elde edilen bibliyografik bilgiler aracılığıyla sayısal verileri kullanan bibliyometrik analiz, yazarlar, kurumlar ve ülkelerle ilgili atıf analizi, atıf grafikleri ve anahtar kelime grafikleri gibi sonuçları analiz etmek ve sunmak için bir yöntem olarak ortaya çıkar. Bibliyometrik analiz, belirli bir konu üzerinde akademik yayın üreten araştırmacıların etkisini ve araştırmacılar arasındaki etkileşimin boyutlarını ortaya çıkarmayı amaçlar. Bibliyometrik analiz, belirli bir alan veya disiplin içindeki kalıpları ve eğilimleri daha iyi anlamak için bibliyografik verilerin sistematik olarak incelenmesini içeren nicel bir araştırma yöntemidir. Bu yöntem, araştırmacıların araştırma alanındaki önemli yazarları, etkili yayınları ve ortaya çıkan konuları belirlemesini sağlar. Bibliyometrik analiz, bilimsel veri tabanlarından, dergilerden, konferans bildirilerinden ve diğer ilgili yayınlardan bibliyografik verilerin toplanmasını ve analiz edilmesini içerir. Genel olarak, bibliyometrik analiz, atıf sayıları, ortak yazarlık ağları,

yayın kalıpları ve anahtar kelime analizi gibi temel metriklerin yazılım araçları veya platformları aracılığıyla belirlenmesine olanak tanır. Bibliyometrik analiz çeşitli araçlar ve teknikler aracılığıyla kolaylaştırılabilir. Bu araçlar, ilgili bilimsel yayınların toplanmasına, düzenlenmesine ve analiz edilmesine yardımcı olur. Bu araçlardan bazıları şunlardır: Scopus, Web of Science, Google Scholar.

Metodoloji

Bu çalışmada, 2012-2024 yılları arasında Web of Science veri tabanında "Akıllı şehirler", "Yapay Zeka" ve "Nesnelerin İnterneti" anahtar sözcükleri kullanılarak yapılan bibliyometrik analiz sonucunda toplam 1375 sonuç elde edilmiştir. Çalışma 24 Ekim 2024 tarihinde yapılmıştır. Konuyla ilgili 2012 yılı ve 2024 yılları arasında yayınlanan çalışmalar ele alınmıştır. 1375 sonuç arasında literatürde "Akıllı şehirler", "Nesnelerin İnterneti" "Yapay Zeka" kavramlarıyla ilgili 915 makale, 259 işlemlem gören makale, 14 kitap bölümü, 12 editoryal materyal, 154 incelemede olan makale, 4 çekilen yayın, 21 erken erişim yayını, ve 1 yeni gelen özet olduğu tespit edilmiştir. VOSviewer programını kullanarak akıllı şehir kavramı üzerine akademik araştırmalar araştırılmış ve haritalanmıştır. Web of Sciences veri tabanından alınan veriler, yazar, atıf, dergi, ülke, organizasyon ve anahtar sözcükler temelinde analiz edilmiştir.

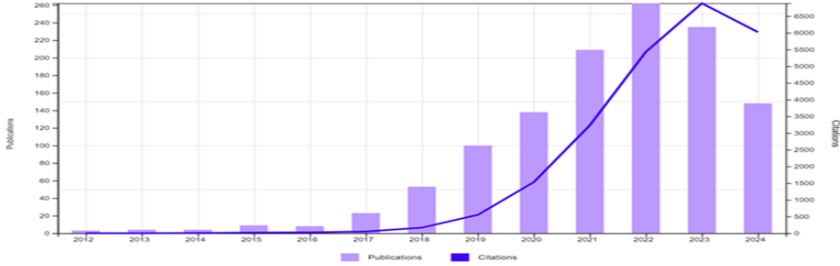
Nicel tekniklerin kullanılması, söz konusu araştırma alanındaki mevcut durumun ve gelişim eğilimlerinin tanımlanmasına olanak tanır. Sonuçlar, belirli bir dönemdeki ana araştırma yönleri, eğilimler ve yayın sayısındaki değişiklikler hakkında fikir verir.

Araştırma Sonuçları

Yıllara Göre Dağılım

Yayınların 2012-2024 veri aralığına dahil edildiği Şekil 1’de, sürekli artışlar ve azalışlarla birlikte süresiz bir evrim

gözlemlenmektedir. En üretken yıl 2022 ile en düşük yayın sayısına sahip yıl 2012 olduğu görülmektedir.



Şekil 1. 2012-2024 yılları arasındaki yayın ve atıfları göstermektedir.

Tablo 1. Yıllara Göre Yayın Sayısını Göstermektedir.

20	20	20	20	20	20	20	20	20	20	20	20	20
24	23	22	21	20	19	18	17	16	15	14	13	12
19	27	30	23	15	10	54	25	11	9	4	4	3
5	6	0	1	7	6							

Kaynak: Yazar tarafından düzenlenmiştir.

Ortak Yazar Analizi (Co-Authorship with Authors)

Ortak yazar analizi, yazarlar arasındaki iş birliği ağlarını tespit etmek için kullanılmaktadır (Kurnaz, 2021; Hırlak, 2024).

Ortak Yazarlı Yazar Analizi

Bu analizde en az bir yayını ve en az bir atfı bulunan yazarların listelenmesi baz alınmıştır. Bu kriter seçimi ile 187 yazardan 143'i incelemeye dahil edilmiştir. Daha büyük daireler ve harita etiketleri daha büyük önemi temsil etmektedir. Benzer renklere sahip olanlar aynı kümeye aittir (van Eck ve Waltman, 2010). Şekil 2'de yazarların konu hakkındaki yayın sayısına göre yoğunluğu ve yazarlar arası iş birliği ağı görülmektedir.

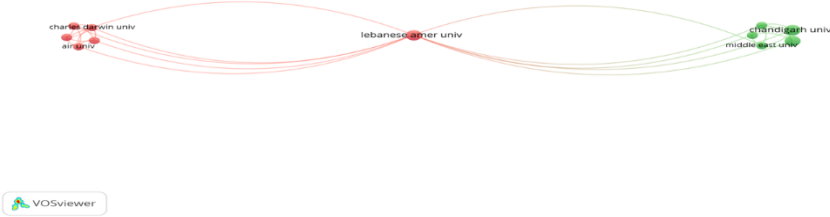
Şekil 2. Ortak Yazar Analizi Ağ Haritası

Yazarların ilişki durumunun iki alt kümede toplandığı görülmektedir. Her bir kümedeki yazarların birbiri ile çalışmaları bulunmaktadır. Bununla beraber bu analizle bu konu hakkında en çok yayın yapan ve en çok atıf alan yazarlar görülmektedir. Sayısal değer olarak bakılacak olduğunda konu hakkında en çok atıf alan ilk 10 yazar Tablo 2’de gösterilmektedir.

Tablo 2. Ortak Yazarlı Ağ Kümesi

Küme 1 (Kırmızı)	Küme 2 (Yeşil)
Cho, gi hwan	Jo, jeong hoon
Ra, in-ho	Park, jong hyuk
Shojafar, mohammad	Sharma, pradip kumar
Sing, saurabh	Sicato, jose costa sapalo
Yoon, byungun	

Ortak Yazarlı Organizasyon Analizi (Co - authorship with Organizations)



Şekil 3. Ortak Yazarlı Organizasyon Analizi Haritası.

Organizasyonda en az 1 makale ve 1 atıf olsun, 133 organizasyondan 102 si bu eşiği gerçekleştiriyor. İki kümeden oluşuyor. Kırmızı ve yeşil kümeler Tablo 3’de gösterilmiştir. Tablo 4’de organizasyonların makale ve atıf sayıları görülmektedir. lebanese amer univ, 2 yayın, 41 atıf ile birinci; birla inst technol &

sci pilani 2 yayın, 32 atıf ile ikinci; univ amer 2 yayın, 48 atıf ile üçüncü sırada görülmektedir.

Tablo 3. Ortak Yazarlı Organizasyon Ağ Kümesi

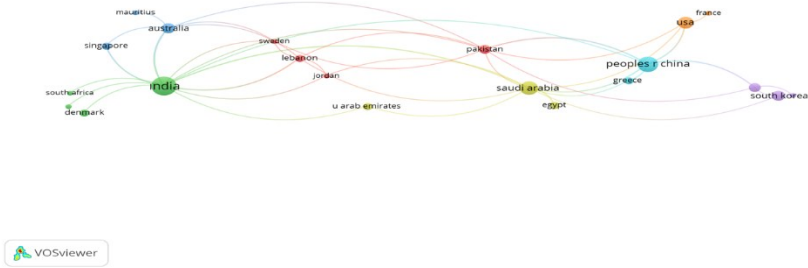
Küme 1 (Kırmızı)	Küme 2 (yeşil)
air univ	chandigarh univ
charles Darwin univ	chitkara univ
lebanese amer univ	lovely profess uni
linnaeus univ	middle east univ
symbiosis int deemed univ	princess nourah bint abdul
vellore ins technol	

Tablo 4. Ortak Yazarlı Organizasyonda Makale ve Atıf sayısı Ağ Kümesi

Organizasyon	Makale	Atıf	Etki gücü
lebanese amer univ	2	41	10
birla inst technol&sci pilani	2	32	9
univ amer	2	48	7
beijing univ technol	1	19	6
ccis prince sultan univ	1	19	6
chandigarh univ	2	1	6
childrens natl hosp	1	19	6
chitkara univ	2	1	6
csir cso	1	30	6
dıt univ	1	30	6
gulzar grp inst	1	30	6
guru nanak dev engn coll	1	30	6
king khalid univ	1	30	6
manipal acad higher educ	1	30	6
univ cent punjab	1	19	6
univ educ	1	19	6
univ engn& technol taxila	1	19	6
air univ	1	40	5
chaes Darwin univ	1	40	5
linnaeus univ	1	40	5
lovely profess univ	1	1	5
Middle east univ	1	1	5
princess nourah bint abdurrahmahman univ	1	1	5
sybiosis int deemed bello	1	40	5
univ andress bello	1	11	5
univ bernardo ohigging	1	11	5
univ cent chile	1	11	5
univ Santiago chile	1	11	5
vellore inst technol	1	40	5
dogguk univ	1	237	4

Ortak Yazarlı Ülke Atıf Analizi (Co - authorship with Countries)

Ülke atıf analizi, konu ile ilgili yayınların ülkelere göre dağılımını ve aldıkları atıf sayıları arasındaki etkileşimi ortaya koymaktadır. Ülkenin en az 1 yayını ve en az 1 atıfı olsun. 39 ülkeden 32 ülke bu eşik değerini sağladığı ve 7 kümeden oluştuğu şekil 4’de görülmektedir. Atıf sayılarına dayalı sonuçlara bakıldığında, australia 4 yayın ve 483 atıfı ile ilk sırada, mauritius 1 yayın ve 440 atıfı ile ikinci sırada ve South korea 4 yayın ve 271 atıfı ile üçüncü sırada yer aldığı, makale sayılarına bakıldığında, india 13 yayın ve 217 atıfı ile ilk sırada, peoples r china 9 yayın ve 179 atıfı ile ikinci sırada ve saudi arabia 7 yayın ve 78 atıfı ile üçüncü sırada yer aldığı görülmektedir. Ülke atıf analizinden elde edilen etkileşim grafiği aşağıda Şekil 4’de sunulmuştur. En büyük halkaya sahip ülke en çok yayını olan ülkedir. Tablo 4’de aynı kümelerde yer alan ülkeler bulunmaktadır.

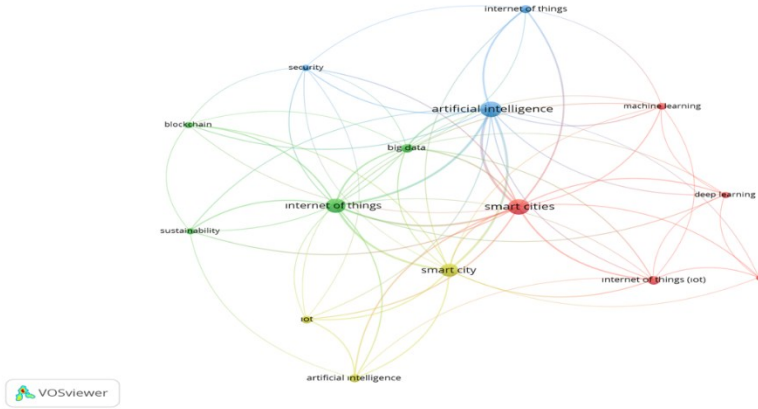


Şekil 4. Ülke Atıf Analizi ağ Haritası

Tablo 5. Ülke Atıf Analizi Ağ Kümesi

Küme1	Küme 2	Küme 3	Küme 4	Küme 5	Küme 6	Küme 7
Kırmızı	Yeşil	Mavi	Açık yeşil	Mor	Mavi	Portakal
Jordan	Denmark	Australia	Egypt	England	Greece	France
Lebanon	India	Mauritius	Saudi	Scotland	Peoples	Usa
Pakistan	South	singapore	arabia	South	r chine	
sweden	Africa		U arab	korea		
	Thailand		emiratet			

Eş Anlamlı Kelime Analizi (Co-Occurrence with Author Keywords)



Şekil 5. Eş Anlamlı Kelime Analizi

Ortak kelime ağı, anahtar kelimelerin birlikte kullanım durumları doğrultusunda düğümleri ve bağlantıları göstermektedir. Renkleri aynı olan düğümler, kümeleri; düğüm boyutları ise kelime sıklıklarını ifade etmektedir. Beraber oluşma sıklığı, aynı kümedeki düğümlerin arasındaki mesafeden ve aralarındaki bağlantıların kalınlığından fark edilebilmektedir (Atabay vd., 2019; Hırlak, 2024).

Akıllı şehir, nesnelerin interneti, yapay zeka konusunda en çok kullanılan anahtar kelimelerin belirlenmesi ve bu değişkenler arasındaki ilişkilerin tespit edilmesi amacı ile anahtar kelime analizi yapıldı. En az 3 anahtar kelime bulunma koşulu baz alındığında, 171

anaktar kelimededen 15 anaktar kelime bu koşulu sağladı. Şekil 5’de birbiriyle ilgili kelimeler 4 küme oluşturmuştur. Tablo 6’da en çok tekrar eden kelimenin "artificial intelligence" olduğu görülmektedir.

Tablo 6. Eş Anlamlı Kelime Analizi küme dağılımı

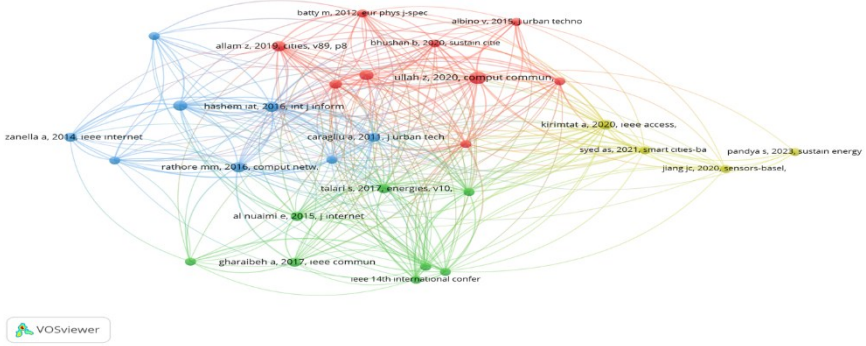
Anahtar Kelime	Tekrar Eden Anahtar Kelimeler	Etki gücü
artificial intelligence	25	64
smart cities	25	62
internet of things	22	54
smart city	19	42
big data	8	28
internet of things(tot)	8	18
internet of things	7	16
Iot	6	16
artificial intelligence	7	15
Blockchain	4	13
Security	4	13
Sustainability	4	13
machine learning	5	12
deep learning	4	11
artificial intelligence(ar)	3	7

Ortak Atıf Analizi (Co-citation Analysis)

Ortak atıf analizi (Co-citation analysis), belirli yayınların ya da yazarların başka çalışmalarda birlikte ne sıklıkla atıf aldığını incelemektedir. Bu analiz, literatürdeki belirli araştırma alanlarını, temaları veya entelektüel yapıları ortaya çıkarmak için kullanılmaktadır.

Atıf Yapılan Referansların Ortak Atıf Analizi (Co-Citation with Cited References)

Orijinal örnekleme 1375 yayından en az 4 kez atıf yapılan 29 yayına düşürüldü. Bu 29 en çok atıf yapılan referansa dayanarak, bu çalışma platform araştırma alanı içinde ortak atıf analizi yoluyla ağı oluşturdu. Sonuçlar 29 referansın dört kümeye ayrıldığını, her rengin bir kümeyi temsil ettiğini gösteriyor. Şekil 6 'da atıf yapılan referansların ortak atıf analizi sonuçları görülmektedir.



Şekil 6. Atıf Yapılan Referansların Ortak Atıf Analizi Haritası (Co-Citation with Cited References)

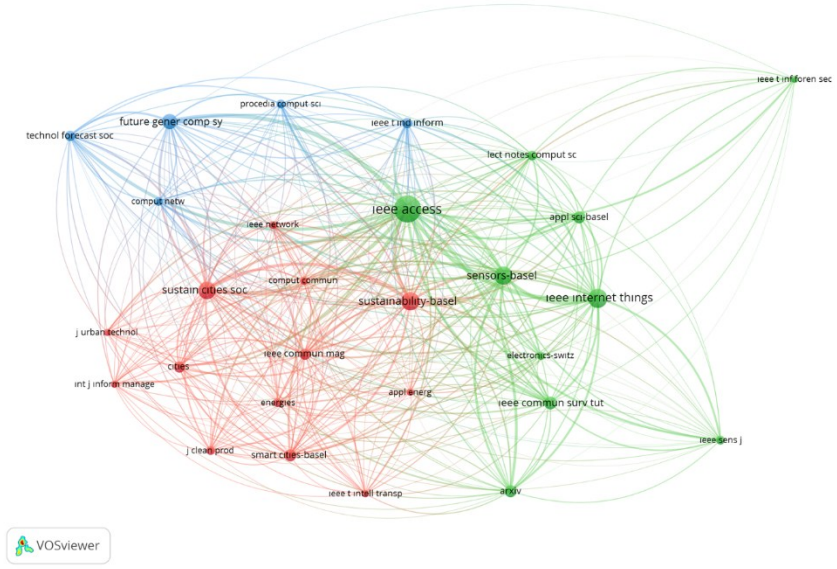
Tablo 7'de ilk on sırada yer alan yazarlardan, en yüksek ortak atıfa sahip yazarlar atıf sayısına göre sıralanmıştır.

Tablo 7. Atıf Yapılan Referansların Ortak Atıf Küme Analizi

Atıf Yapılan Kaynaklar	Atıf	Etki gücü
Hashem iat, 2016, int j inform manage, v36, p738, doi 1...	7	67
Ullah z, 2020, comput commun, v154, p313, doi10.1016	10	61
Kirimtat a, 2020, iee Access, v8, p86448, doi 10.1109/ac...	6	52
Allam z, 2019, cities, v89, p80, doi 10.1016/j.cities.2019.0...	7	48
Caragliu a, 2011, j urban technol, v18, p65, doi 10.1080/...	5	45
Neirotti p, 2014, cities, v38, p25, doi 10.1016/j.cities.2013...	7	45
Osman ams, 2019, future gener comp sy, v91, p620, doi...	5	45
Bhusman b, 2020, sustain cities soc, v61, doi 10.1016/j.s...	4	44
Xie jf, 2019, IEEE commun surv tut, v21, p2794, doi 10.1...	4	44
Silva bn, 2018, sustain cities soc, v38, p697, doi 10.1016/...	7	43

Atıf Yapılan Kaynağın Ortak Atıf Analizi (Co-Citation with Cited Sources)

Ortak atıf analizi, belirli bir araştırma alanında yürütülen çalışmalar arasındaki ilişkileri ve etkileşimleri inceleyen bir yöntemdir. Bu yöntem, bir makalenin referanslarını analiz ederek makalenin hangi diğer çalışmalara atıf yaptığını değerlendirir ve bu atıfların doğasını değerlendirir. Kaynak atıf analizi, konu hakkında en çok atıf alan yayınlar ile aldıkları atıf sayısı arasındaki ilişkiyi ortaya koyar. Kaynak atıf analizi yoluyla, 'Akıllı şehir', 'Nesnelerin İnterneti' ve 'Yapay zeka' konularıyla ilgili, bir kaynağın en az atıf alma sayısı 15 olarak seçilerek 2220 kaynaktan 28 kaynak bu koşulu sağlayarak etkileşim grafiğinde üç bağlantı kümesi elde edilmiştir. Bu bağlamda, en çok atıf alan kaynağın 200 atıf ile 'IEEE access' olduğu, bunu 104 atıf ile 'IEEE internet things' izlediği görülmüştür. Şekil 7'de Atıf Yapılan Kaynağın Ortak Atıf Analizi sonuçları sunulmuştur.



Şekil 7. Atıf Yapılan Kaynağın Ortak Atıf Analizi Haritası

Tablo 8’de en yüksek atıf alan 14 kaynak sırasıyla gösterilmiştir.

Tablo 8. Atıf Yapılan Kaynağın Ortak Atıf Analizi

Kaynak	Atıflar	Kaynak	Atıflar
IEEE Access	200	lect notes comput sc	26
IEEE internet things	104	j clean prod	23
sensors-basel	94	procedia comput sci	22
sustainability-basel	89	comput netw	22
sustain cities soc	82	IEEE sens j	22
future gener comp sy	63	Energies	21
IEEE commun surv tut	45	IEEE network	19
apply sci-basel	42	comput commun	18
Arxiv	39	electronics- switz	17
smart cities-basel	38	int j inform manage	16
IEEE commun mag	37	IEEE t intell transp	16
Cities	35	IEEE t inf foren sec	16
technol forecast soc	30	j urban technol	15
IEEE t ind inform	30	appl energ	15

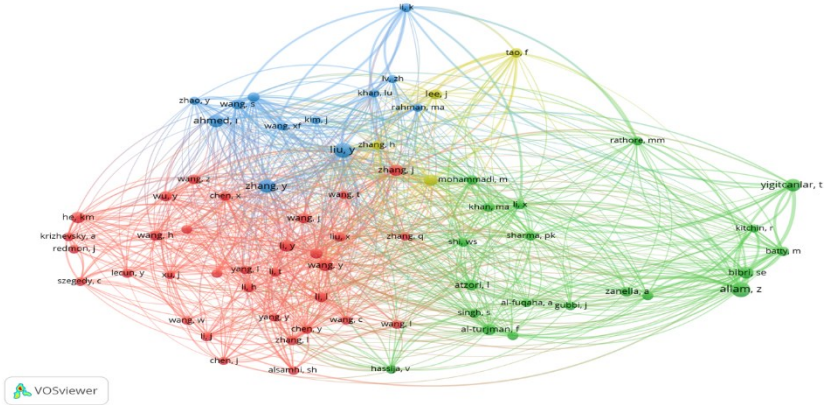
Ortak Atıf Analizi (Co-Citation with Cited Authors)

Ortak atıf analizinde 4 kümede toplanma, akıl şehirlerde nesnelerin interneti, yapay zekâ konusunun 4 ana araştırma alanı veya tema olduğunu ve bu alanların her birinin belirli çalışmalar ve yazarlar etrafında şekillendiğini göstermektedir. Bu alanlar anahtar kelime analizi başlığı altında daha detaylı gösterilmektedir.

Bir yazarın en az 30 atıfı olsun. 34914 yazardan, 69 yazar bu eşik değerini sağladı. 69 yazar 4 küme oluşturdu. Bununla beraber bu ağların 4 farklı kümede toplandığı görülmüştür. Ortak atıf analizinde 4 kümede toplanma, 4 ana araştırma alanı veya tema

olduğunu ve bu alanların her birinin belirli çalışmalar ve yazarlar etrafında şekillendiğini göstermektedir. Şekil 8’de en yüksek atıf alan iki yazar; Liu, y ve Allam, z, ortak atıf konusunda etrafında yoğunlaşmış yazarlar olarak görülmektedirler.

Analiz, aynı makalelerde birlikte görünen yazarların aldığı atıfları incelemeyi ve ortak yazarlar tarafından yürütülen işbirlikli araştırmanın etkisini değerlendirmeyi ve araştırma ağlarını belirlemeyi amaçlamıştır. Bu analizde, aynı makalede birlikte görünen yazarlara yapılan atıflar, aynı makaleye atıfta bulunan diğer makalelerde belirlenmiştir. Analizin sonuçları şekil 8’de aşağıda sunulmuştur.



Şekil 8. Yazarların ortak atıf analizinin dağılımı (Küme yoğunluklarının dağılımı)

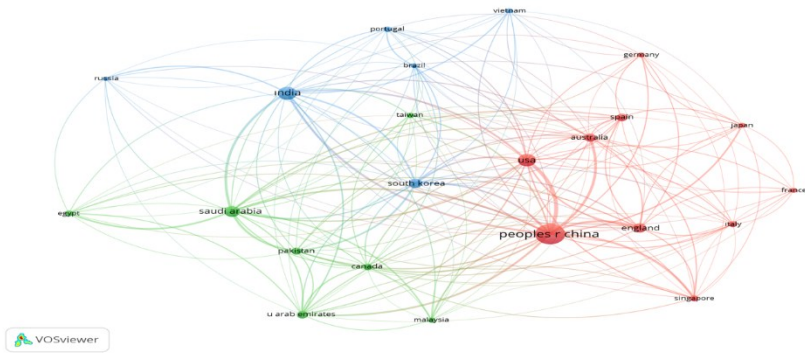
Tablo 9’de En yüksek atıf alan 10 yazar yer almaktadır.

Tablo 9. Yazarların Ortak Atıf Analizi

Yazar	Atıf
Liu, y	102
Allam, z	97
Zhang, y	76
Ahmed, ı	67
Yiğitcanlar, t	66
Wang, y	65
Li, y	63
Bibri, se	61
Zhang, j	57
Al-turjman, f	55

Ülke Atıf Analizi (Citation with Countries)

Ülke atıf analizi, konu ile ilgili yayınların ülkelere göre dağılımını ve aldıkları atıf sayıları arasındaki etkileşimi ortaya koymaktadır. Ülkenin en az yayın sayısı 21 ve en az atıf sayısı 20 olsun. 100 ülkeden 23 ülke bu koşulu sağladı. Atıf sayılarına dayalı sonuçlara bakıldığında, Peoples R China 430 yayın ve 10220 atıfıla ilk sırada, USA 158 yayın ve 6111 atıfıla ikinci sırada ve İngiltere 93 yayın ve 4123 atıfıla üçüncü sırada yer aldığı görülmektedir. Ülke atıf analizinden elde edilen etkileşim grafiği şekil 9'da sunulmuştur.



Şekil 9. Ülkeye Göre Atıfların Ağ Görselleştirilmesi

Bilim alanında lider ülkeleri ve bu ulusların yıllar içindeki evrimini belirlemek için coğrafi bir analiz yapıldı. Ortak yazarlı ülkeler Tablo 10 'da sunulan bilimsel üretimin coğrafi dağılımına ilişkin verileri göstermektedir. Bibiriyle ilişkili ülkeler 3 kümeden oluşmaktadır. Tablo 11'de ülkelere göre yayın ve atıf sayısı yer almaktadır. Çin Halk Cumhuriyeti, USA, İngiltere, Hindistan şeklinde dağılım göstermektedir.

Tablo 10. Ülkeye Göre Atıfların Kümesi

Küme 1	Küme 2	Küme 3
Australia, England, France, Germany, italy, japan, poeple r china, singapore, usa	Canada, malaysia, saudi arabia, arab emirates	egypt, Pakistan, taiwan, u South korea, vietnam india, russia,

Tablo 11. Ülkelere Göre Küme Dağılımı

Ülke	Yayın	Atıf
People r China	430	10220
USA	158	6111
England	93	4123
Australia	67	3783
India	171	3019
Singapore	41	2756
Soulth Korea	95	2647
Saudi Arabia	129	2071
U Arap Emirates	65	1856
Canada	49	1385

Sonuç ve Değerlendirme

Bu çalışma literatür taramasından elde edilen çalışmalarını özetler. WoS'ta elde edilen performans analizini kullanılarak, bu çalışma en çok atıf alan makaleler, en etkin yazarlar, en etkili dergileri, en etkili kurumları ve en çok yayın yapan ülkeleri en çok atıf alan referanslar belirlendi.

Bu çalışma, 2012-2024' yılları arasında web of science verilerini kullanarak, akıllı şehirler için literatürde yayınlanan arařtırmalara genel bir bakıř saęlamayı ve řu anda yürütölmekte olan arařtırmanın kapsamlı bir göröntüsünü sunmayı amaçlamaktadır. Arařtırmacılar, arařtırmaya göre akıllı şehirler için Nesnelere İnterneti ve yapay zeka çalışmalarının 2016-2024 yılları arasında büyüme gösteren, hızla yükselen bir arařtırma konusu olduęunu ortaya koymaktadır. Çin, Amerika Birleřik Devletleri, İngiltere Avustralya vs. olmak üzere ölkelerden çok çeřitli bilim insanları, özellikle Çin ve Amerika Birleřik Devletleri arasında uluslararası iř birlięi için zemin oluřturduęu görölmektedir. Bibliyometrik analiz kullanımıyla, arařtırma ortamı, önemli makalelerin, kaynakların ve yazarların incelenmesi de dahil olmak üzere çeřitli düzeylerde arařtırıldı. Yazarlar, nesnelere interneti, yapay zeka, makine öğrenimi vs. akıllı şehirler üzerine yapılan arařtırmaların akıllı şehirlerin sorunlarına yardımcı olmak için büyük bir potansiyele sahip olduęu sonucuna varıldı.

KAYNAKÇA

Atabay, E., Çizel, B., & Ajanovic, E. (2019). Akıllı şehir arařtırmalarının R programı ile bibliometrik analizi. O. Emir (Ed.), Akıllı řehirler, 20. Ulusal Turizm Kongresi, Anadolu Üniversitesi Basımevi, 3, 1130-1137.

Hırlak, B. (2024). Yönetim alanındaki whistleblowing arařtırmalarının bibliyometrik analizi. R&S – Research Studies Anatolia Journal, 7(2), 154-185. <https://doi.org/10.33723/rs.1451312>

Kurnaz, A. (2021). Etnosentrizm ile ilgili çalışmaların bibliyometrik analizi. Beykoz Akademi Dergisi, 9(2), 98-118. <https://doi.org/10.14514/BYK.m.26515393.2021.9/2.98-118>

van Eck, N.J., Waltman, L. (2010). Software survey: VOSviewer, a computer program for bibliometric mapping. Scientometrics, 84,523–538. <https://doi.org/10.1007/S11192-009-0146-3>

Wang J., Jiang C., Zhang K., Quek T.Q., Ren Y., Hanzo L. (2018). Vehicular sensing networks in a smart city: Principles, technologies and applications. IEEE Wirel. Commun. 25:122–132. doi: 10.1109/MWC.2017.1600275.

Winkowska, J., Szpilko, D., & Pejić, S. (2019). Smart cityconcept in the light of the literature review. Engineering Management in Production and Services, 11, 70-86. doi: 10.2478/emj-2019-0012 World’s population increasingly urban with more than half living in urban areas | UN DESA | United Nations Department of Economic and Social Affairs. (2014). Retrieved April 23, 2017, from

<http://www.un.org/en/development/desa/news/population/worldurbanization-prospects-2014.htm>

