ARALIK 2025

İstatistik Alanında Uluslararası Derleme, Araştırma ve Çalışmalar

EDİTÖR
DOÇ. DR. FÜSUN YALÇIN

SERÜVEN YAYINEVİ

YERALTISULARI MEVZİİATİ VE İYİGİİ AMASI

PROF. DR. AHMET AYDIN

Genel Yayın Yönetmeni / Editor in Chief • C. Cansın Selin Temana

Kapak & İç Tasarım / Cover & Interior Design • Serüven Yayınevi

Birinci Basım / First Edition • © Aralık 2025

ISBN • 978-625-8671-19-3

Serüven Yayınevi / Serüven Publishing
Türkiye Adres / Turkey Address: Kızılay Mah. Fevzi Çakmak 1. Sokak
Ümit Apt No: 22/A Çankaya/ANKARA
Telefon / Phone: 05437675765
web: www.seruvenyayinevi.com
e-mail: seruvenyayinevi@gmail.com

Baskı & Cilt / Printing & Volume
Sertifika / Certificate No: 47083

İSTATİSTİK ALANINDA ULUSLARARASI DERLEME, ARAŞTIRMA VE ÇALIŞMALAR

- ARALIK 2025 -

EDİTÖR
DOÇ. DR. FÜSUN YALÇIN

İÇİNDEKİLER

Bölüm 1

RASTGELE TEK GİZLİ KATMANLI İLERİ BESLEMELİ YAPAY SİNİR AĞLARINDA META-SEZGİSEL OPTİMİZASYON ALGORİTMALARI ÜZERİNE İNCELEME

Çağatay BAL[1]

1 Arş. Gör. Dr. Muğla Sıtkı Koçman Üniversitesi, Fen Fakültesi, İstatistik Bölümü, Muğla/Türkiye, cagataybal@mu.edu.tr, https://orcid.org/0000-0002-7823-2712

1. Giriş

Yapay sinir ağı (YSA), makine öğrenimi (ML) ve yapay zekanın (AI) kilit tekniği olduğu yaygın olarak iddia edilmektedir [1,2]. Fonksiyon yaklaştırma [3,4], özellik seçimi [5,6], örüntü tanıma [7-11] ve daha karmaşık görevlerde [12,13] başarıyla kullanılmıştır. Tipik olarak gradyan tabanlı öğrenme algoritması ile gerçekleştirilen ileri beslemeli YSA, doğrusal veya doğrusal olmayan eşlemeler üzerindeki evrensel yaklaştırma yeteneği nedeniyle gelecek vadeden bir teknoloji haline gelmiştir [14-17]. Yine de, geri yayılım (BP) algoritması gibi gradyan tabanlı öğrenme yöntemleri, binlerce iterasyonla gradyan bilgisinin hesaplanması nedeniyle çok zaman alıcıdır ve genellikle yerel optimuma yakınsama sorunu yaşarlar [18-20].

Geleneksel gradyan tabanlı öğrenme algoritmalarını geliştirmek için, ağın yerel optimuma takılmasını önlemek amacıyla gradyan yönelimindeki keskin değişimi yumuşatabilen momentum faktörü tanıtılmıştır [21]. Ayrıca, BP algoritmasından daha iyi genelleme performansı ile daha hızlı yakınsayan eşlenik gradyan algoritması [22], Newton yöntemi [23], stokastik gradyan inişi (SGD) [24] ve uyarlanabilir moment tahmini (Adam) [25] de ileri beslemeli YSA'da kullanılmıştır. Ancak, insan sezgisiyle yorumlanamadığı için yukarıdaki yöntemlerde sinir ağı "kara kutu" olarak görülmektedir [26]. Bu nedenle, temel olarak iki alt modelden oluşan genelleştirilmiş kısıtlı bir sinir ağı (GCNN) modeli geliştirilmiştir [26]. Biri, hedef fonksiyonun bilinmeyen kısmına yaklaşmak için standart sinir ağı tekniği ile oluşturulur. Diğeri ise tüm modele genelleştirilmiş kısıtlamalar uygulamak için kısmen bilinen ilişkilerden oluşturulur [27-32]. Ek olarak, GCNN çalışmaları, ön bilgiyi sinir ağlarının tasarımına dahil etmek [33-35] ve sinir ağlarına gömülü bilgiyi çıkarmak olmak üzere iki tür stratejiye ayrılmıştır. Önceki deneyimlerden veya verilerden elde edilen ön bilgiyi dayatmak, ağın imkansız tahminler yapmasını engelleyebilir ve genelleme performansını artırabilir [36-38]. Dahası, bağlam bilgisi problem alanından temsil biçimine kadar heterojen olduğundan, ön bilgiyi ileri beslemeli YSA'ya gömmek zor bir görevdir [26].

Ayrıca, gradyan tabanlı öğrenme yöntemlerinin ve varyantlarının eğitim sürecini hızlandırmak için rastgele tek gizli katmanlı ileri beslemeli sinir ağı (RSLFN) geliştirilmiştir. Bir RSLFN sınıfı olarak, tek gizli katmanlı ileri beslemeli sinir ağını (SLFN) eğitmek için rastgele vektör fonksiyonel bağlantı ağları (RVFL), [39]'da önerilmiştir. RVFL'de, orijinal girdiler ilk olarak geliştirme düğümleri tarafından doğrusal olmayan gizli çıktılara dönüştürülür ve daha sonra tüm gizli çıktılar ve orijinal girdiler birleştirilerek çıktı nöronlarına beslenir (bkz. Şekil 1a) [40]. Bir RVFL, girdi ağırlıklarının önceden bağımsız olarak seçilen rastgele değerlerle sınırlandırıldığı yarı rastgele bir SLFN uygulaması olarak kabul edilebilir. Kalan parametreler yalnızca eşlenik BP ile ayarlanan uyarlanabilir parametreler olarak seçilir [41,42]. Böylece RVFL basit ve hızlı bir öğrenme sürecine sahiptir ve sınırlı

iterasyonlarda optimal çözüme yakınsaması garanti edilir [43-45]. Ek olarak, Schmidt ve ark. [46] rastgele ağırlıklara sahip bir SLFN modeli (RWSLFN) önermiştir. Herhangi bir RWSLFN için, girdi ağırlıkları ve gizli önyargılar (bias) [-1,1] aralığındaki tekdüze bir dağılımdan rastgele üretilir ve çıktı ağırlıkları iteratif öğrenme kuralı kullanılarak belirlenir [46]. Öğrenme kuralı, RWSLFN'in yapısı sabitlendiğinde yalnızca gizli katman ile çıktı katmanı arasındaki ağırlıkların öğrenilmesi gerektiği gerçeğine yol açar. Ayrıca, RWSLFN'in girdiler ve çıktılar arasında doğrudan bağlantıları yoktur (bkz. Şekil 1b), bu da RVFL'in yapısını basitleştirir ve hesaplama maliyetini düşürür [47].

ML'deki birçok optimizasyon problemi, kapalı formda sunulabilen iteratif olmayan yaklaşımlarla ele alınabilir. İteratif yöntemlerle karşılaştırıldığında, iteratif olmayan yaklaşımlar hesaplama maliyetini önemli ölçüde azaltır ancak yine de tatmin edici sonuçlara sahiptir [48,49]. Bu nedenle, geleneksel gradyan tabanlı YSA ve iteratif RSLFN'in kusurlarını aşmak için iteratif olmayan öğrenme yöntemine sahip RSLFN önerilmiştir. İteratif olmayan öğrenme yöntemine sahip RSLFN'in çıktı ağırlıkları Moore-Penrose genelleştirilmiş tersi yoluyla elde edilebilirken, girdi ağırlıkları ve gizli önyargılar başlangıçta rastgele başlatılır [50]. İteratif olmayan RSLFN genel uygulanabilirliğe ve olumlu performansa sahip olduğundan, metin analizi [51], görüntü sınıflandırma [52] ve doğal dil işleme [53] gibi uygulamalarındaki artış hem akademide hem de endüstride büyük ilgi görmüştür.

Meta-sezgisel algoritma normalde iteratif üretim süreci yoluyla yeterince iyi bir çözüme yakınsar [54,55]. Meta-sezgisel yaklaşımların anahtarları arama uzayını keşfetmek (exploration) ve sömürmektir (exploitation). Evrimsel algoritmalar ve sürü zekası yöntemlerini içeren en popüler meta-sezgiseller, YSA'yı optimize etmek için başarıyla uygulanmıştır [56-58]. Bunlar genellikle YSA parametrelerini bir optimizasyon modeline formüle eder ve ardından daha iyi bir YSA oluşturan ideale yakın bir çözüm sağlama eğilimindedir [59-62]. Bu konulardaki son incelemeler [63-65]'te görülebilir. Meta-sezgisel algoritmalar, optimal parametreleri, en kompakt ağ mimarisini vb. arayarak RSLFN'in eksikliklerinin üstesinden gelmek için de mevcuttur.

2. Ön Bilgiler

Bu bölüm ilk olarak RSLFN'in temel öğrenme kurallarını, özellikle basitleştirilmiş bir versiyon olarak kabul edilen iteratif olmayan RSLFN'i tanıtmaktadır. Ardından, genetik algoritma (GA) [66], diferansiyel gelişim (DE) [67] ve parçacık sürü optimizasyonu (PSO) [68] dahil olmak üzere en yaygın kullanılan meta-sezgisel yaklaşımlar verilmektedir. Son olarak, çok amaçlı optimizasyon ve Pareto optimalliği kısaca açıklanmaktadır.

2.1. İteratif olmayan öğrenme ile RSLFN

N adet keyfi farklı örnek $(x_i, \quad t_i)$ için, burada $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in R^n$ ve $t_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^\tau \in R^m$ dir. $\widetilde{N}$ gizli nöronlu ve aktivasyon fonksiyonu $g(\cdot)$ olan standart bir SLFN, bu N örneğe sıfır hata ile yaklaşabilir, bu şu anlama gelir:

$$\sum_{j=1}^{\widetilde{N}} \beta_j \, g_j(x_i) = \sum_{j=1}^{\widetilde{N}} \beta_j \, g\left(w_j \cdot x_i + b_j\right) = t_i, \; i = 1.2 \ldots \ldots N. \qquad (1)$$

Burada $\beta_j = [\beta_{j1}, \beta_{j2}, \ldots, \beta_{jm}]^\tau$, tüm çıktı nöronlarını ve j. gizli nöronu bağlayan çıktı ağırlık vektörüdür, $w_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$, j. gizli nöronu ve tüm girdi nöronlarını bağlayan girdi ağırlık vektörüdür ve b_j, j. gizli nöronun önyargı (bias) terimidir.

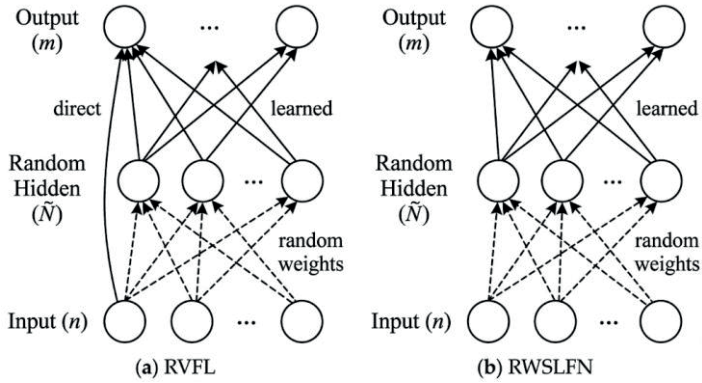
Eşitlik (1) şu şekilde de yazılabilir:

$$H\beta = T \qquad (2)$$

Burada,

$$H(w_1, \ldots, w_{\widetilde{N}}, b_1, \ldots, b_{\widetilde{N}}, x_1, \ldots, x_N) = \left[g\left(w_j \cdot x_i + b_j\right)\right]_{(N \times (\widetilde{N}))}$$

$$\beta = \begin{bmatrix} \beta_1^\dagger \\ \vdots \\ \beta_N^\pi \end{bmatrix}_{N \times m} \text{ ve } T = \begin{bmatrix} t_1^\dagger \\ \vdots \\ t_N^\pi \end{bmatrix}_{N \times m}$$



Şekil 1: RVFL ve RWSLFN mimarileri

Gizli katman ile çıktı katmanını bağlayan ağırlıkların β en küçük kareler çözümü, aşağıdaki gibi yaklaşım hatasını en aza indirerek çözülebilir.

$$\min_\beta ||H\beta - T|| \qquad (3)$$

Rastgele girdi ağırlıkları ve gizli önyargılara sahip RSLFN, doğrusal bir sistem olarak kabul edilir ve en küçük norma sahip çıktı ağırlığı analitik olarak şu şekilde belirlenir:

$$\beta = H^{\dagger}T \tag{4}$$

Burada $H^{\dagger} = (H^T H)^{-1} H^T$, H matrisinin Moore-Penrose genelleştirilmiş tersidir [50]. Daha iyi kararlılık ve genelleme performansı için, maliyet fonksiyonu Eşitlik (3)'te sıklıkla ridge regresyonu olarak da adlandırılan l_2 düzenlileştirmesi (regularization) kullanılır.

$$min_{\beta} ||\beta||_{\mu}^{\sigma_1} + C||H\beta - T||_{v}^{\sigma_2} \tag{5}$$

Burada $\sigma_1 > 0, \sigma_2 > 0, \mu, v = 0{,}1/2{,}1{,}2, \ldots, +\infty, C > 0$, iki hedef arasındaki dengeyi kontrol eden bir parametredir. Ortogonal projeksiyon yöntemleri, iteratif yöntemler ve tekil değer ayrışımı (SVD) dahil ancak bunlarla sınırlı olmamak üzere çıktı ağırlıkları β'yı hesaplamak için çok sayıda verimli yöntem kullanılabilir. $\sigma_1 = \sigma_2 = \mu = v = 2$ ile verimli bir kapalı form çözümü aşağıdaki gibi gösterilmiştir.

$$\beta = \begin{cases} H^T(\frac{I}{C} + HH^T)^{-1}T & if\ N \leq \overline{N} \\ (\frac{I}{C} + H^T H)^{-1} H^T T & if\ N > \overline{N} \end{cases} \tag{6}$$

Verilen yeni bir x örneği ile RSLFN'in karşılık gelen çıktı fonksiyonu şöyledir:

$$f(x) = \begin{cases} g(x)H^T(\frac{I}{C} + HH^T)^{-1}T & if\ N \leq \overline{N} \\ g(x)(\frac{I}{C} + H^T H)^{-1} H^T T & if\ N > \overline{N} \end{cases} \tag{7}$$

2.2. Meta-sezgisel algoritma

2.2.1. Genetik algoritma

GA, büyük bir evrimsel algoritma sınıfına ait stokastik bir arama tekniğidir. GA'da, potansiyel çözümlerden oluşan bir popülasyon, doğal seçilimi simüle ederek sınırlı nesiller boyunca daha iyi popülasyona doğru evrilir [69,70]. Temel operatörler, hepsi doğa evriminden esinlenen mutasyon, çaprazlama ve seçimi kapsar. Bir popülasyondaki bireyler birbirleriyle rekabet eder ve bilgi alışverişinde bulunur. Yeni bireyler ve eski bireyler uygunluk fonksiyonu ile değerlendirilir ve üstün olanların bir sonraki nesil için hayatta kalmasına izin verilir.

2.2.2. Diferansiyel gelişim

Diferansiyel gelişim, Storn ve Price [71] tarafından önerilen güçlü bir stokastik arama tekniğidir. DE, GA tarafından kullanılanlara benzer hesaplama adımlarıyla çalışır. DE'nin temel adımları şu şekilde tanımlanabilir [67]:

Her G neslindeki popülasyonun i. D boyutlu vektörü;

$x_{i.G} = \left(x_{1,i.G}, x_{2,i.G}, \ldots, x_{D,i,G}\right)$, $i = 1{,}2{,}\ldots{,}$NP olarak gösterilsin.

Mutasyon: Her hedef vektör $X_{i,G}$ için, aşağıdakine göre bir mutant vektör üretilir:

$$V_{i,G+1} = X_{r_1.G} + F \cdot \left(X_{r_2.G} - X_{r_3,G}\right) \tag{8}$$

Burada r_1, r_2, r_3 [1, NP] aralığından rastgele ve karşılıklı olarak hariç tutulan indekslerdir ve $F \in [0,2]$'dir. Sabit faktör F, diferansiyel değişimin $\left(X_{r_2,G} - X_{r_3,G}\right)$ güçlendirilmesini kontrol etmek için kullanılır.

Çaprazlama: Deneme vektörü $U_{i,G+1} = \left(u_{1,i,G+1}, u_{2,i,G+1}, \ldots, u_{D,i,G+1}\right)$ aşağıdaki şekilde oluşturulur:

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & if\ rand_{i,j}[0,1] \le Cr \\ x_{j,i,G} & otherwise \end{cases} \tag{9}$$

Veya

$$j = j_{rand} \tag{10}$$

Burada $rand_{i,j}[0,1]$, tekdüze dağılımlı bir rastgele sayının j. değerlendirmesidir, $j_{rand} \in [1, D]$, $U_{i,G+1}$'in $V_{i,G+1}$'den en az bir bileşen almasını sağlayan rastgele seçilmiş bir indekstir, Cr kullanıcı tarafından belirlenen [0,1] aralığındaki çaprazlama sabitidir.

Seçim: Eğer vektör $U_{i,G+1}$, $X_{i,G}$'den daha iyiyse, o zaman $X_{i,G+1}$, $U_{i,G+1}$ olarak ayarlanır. Aksi takdirde, eski değer $X_{i,G}$, $X_{i,G+1}$ olarak tutulur.

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & if\ f\left(U_{i,G+1}\right) \le f\left(X_{i,G}\right) \\ X_{i,G} & otherwise \end{cases} \tag{11}$$

2.2.3. Parçacık sürü optimizasyonu

Parçacık sürü optimizasyonu, Eberhart ve Kennedy [72,73] tarafından önerilen popülasyon tabanlı bir stokastik küresel optimizasyon yöntemidir. PSO ve varyantları, arama uzayı D üzerinde çok sayıda parçacığı başlatarak çalışır. Çoklu iterasyonlar sırasında küresel optimumu (PSO'da küresel en iyi konum olarak anılır) bulmak için, i. parçacık momentum vektörü $v_i = (v_{i1}, v_{i1}, \ldots, v_{iD})$, en iyi konum $P_{ib} = (p_{i1}, p_{i1}, \ldots, p_{iD})$ ve popülasyonun en iyi konumu $P_g = \left(p_{g1}, p_{g1}, \cdots, p_{gD}\right)$'ye göre belirli bir hızla uçar. i. parçacığın mevcut konumu $X_i = (x_{i1}, x_{i1}, \ldots, x_{iD})$ olarak ifade edilir. Temel PSO aşağıdaki gibi gösterilebilir [73]:

$$v_{id}(t+1) = v_{id}(t) + c_1 \times rand(\ \ ) \times [p_{id}(t) - x_{id}(t)] + c_2 \times rand(\ \ ) \times \left[p_{gd}(t) - x_{id}(t)\right] \tag{12}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1)\ 1 < i < NP\ 1 \le d \le D \tag{13}$$

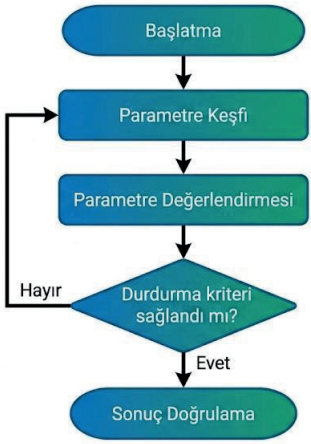
$$P_{ib} = \begin{cases} X_i & if\ f(X_i) \le f(P_{ib}) \\ P_{ib} & otherwise \end{cases} \quad (14)$$

$$P_g = \begin{cases} X_i & if\ f(X_i) \le f(P_g) \\ P_g & otherwise \end{cases} \quad (15)$$

Burada c_1 ve c_2 iki pozitif hızlanma sabitidir, $rand()$ (0,1) aralığında rastgele bir sayıdır.

2.3. Çok amaçlı optimizasyon

Genel bir çok amaçlı optimizasyon problemi şu şekilde ifade edilir:min $F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T$ s.t. $x \in \Omega$(16)Burada $x = [x_1, x_2, \dots, x_n]^T$ karar değişkenleri vektörüdür, $\Omega \in R^m$ uygun bölgedir ve F, m adet amaç fonksiyonundan oluşur.



Şekil 2: Meta-sezgisel ile optimize edilmiş geliştirilmiş RSLFN'in genel çerçevesi

Özellikle, hiçbir uygun çözümün tüm hedefleri en aza indiremediği durum nedeniyle bu hedefler birbirleriyle çelişebilir. x_A, $x_B \in \Omega$ olmak üzere iki karar vektörü düşünüldüğünde, x_A vektörü x_B'ye ($x_A \prec x_B$) ancak ve ancak şu durumlarda baskın gelir (dominate) [74]:

$$\forall i = 1,2, \dots, m\ f_i(x_A) \le f_i(x_B) \wedge \exists j = 1,2, \dots, m.\ f_j(x_A) < f_j(x_B) \quad (17)$$

Dahası, Pareto optimalliği dahil edilir ve bir x^* çözümü, Eşitlik (16) için ancak ve ancak şu durumda Pareto optimal vektör (baskın olmayan çözüm) olarak kabul edilir:

$$\emptyset^* \triangleq \{x^* \in \Omega | \neg \exists x \in \Omega, x \prec x^*\} \quad (18)$$

Böylece, tüm Pareto optimal vektörlerinden oluşan Pareto cephesi şu şekilde tanımlanır:

$$\emptyset: \tilde{F}^* \triangleq \{F(x^*) = (f_1(x^*), f_2(x^*), \dots, f_m(x^*))^T | x^* \in \mathcal{F}^*\} \quad (19)$$

3. Meta-sezgisellerle optimize edilmiş geliştirilmiş RSLFN'ler

RSLFN, geleneksel SLFN'lere kıyasla daha hızlı öğrenme hızına ve daha iyi genelleme performansına sahiptir, ancak tahmin yeteneği rastgele üretilen girdi ağırlıkları ve gizli nöronların sayısıyla yakından ilişkilidir. Şekil 2, meta-sezgisele dayalı geliştirilmiş RSLFN'in genel çerçevesini göstermektedir.

3.1. Girdi ağırlıklarının ve gizli önyargıların optimizasyonu

Meta-sezgisel algoritmaya sahip RSLFN, yüksek olasılıkla optimal olmayan gizli düğümlerden kaçınmak için girdi ağırlıklarını ve gizli önyargıları aramaya odaklanır. RSLFN'in girdi ağırlıkları ve gizli önyargıları öncelikle bir birey olarak D boyutlu bir vektöre kodlanır, burada D toplam girdi ağırlığı ve gizli önyargı sayısıdır. Optimizasyon sürecinde, çok sayıda birey içeren popülasyon meta-sezgisel operatörlerle evrilir ve MP genelleştirilmiş tersi ile RSLFN'ler oluşturmak için kullanılır. Her bireyin uygunluğu genellikle eğitim veya doğrulama setindeki tahmin doğruluğu ile değerlendirilir. Geliştirilmiş RSLFN sonunda meta-sezgisel algoritma tarafından bulunan optimal girdi ağırlıkları ve gizli önyargılar ile oluşturulur ve genellikle en yüksek tahmin yeteneğine sahiptir.

3.1.1. Kodlama şeması

Ağırlık ve önyargı optimizasyonu için iki popüler kodlama şeması ikili kodlama ve gerçek kodlamadır. İkili kodlama, sinir ağının ağırlıklarının ve önyargılarının 0-1 dizileri ile ifade edilmesidir. GA'da kromozomlar olarak da adlandırılan ayrık diziler, meta-sezgisel algoritmanın ikili bir çözüm uzayında arama yapmasını sağlar. Whitley [75] ilk olarak YSA ağırlıklarını eğitmek için ikili kodlamalı GA kullanmayı önermiş ve diğer çalışmalar bu çerçeveyi izlemiştir. Ancak, bir kromozomun en iyi uzunluğunun ne olacağı ve ağırlıkları temsil etmek için kaç bitin yeterli olacağı açık sorulardır [65]. Bu nedenle, ağırlık optimizasyon stratejisi olarak gerçek kodlamayı benimsemenin daha pratik bir yol olduğu öne sürülmektedir.

Gerçek kodlama, kromozomun bileşenlerinin veya diğer birey türlerinin sürekli değerlere eşlenmesini sağlar. YSA eğitimi için, gerçek kodlamalı meta-sezgisel algoritma genellikle gradyan tabanlı öğrenme algoritmasından daha iyi performans gösterir, çünkü küresel optimuma çok daha fazla yaklaşabilir [76,77]. GA, DE veya PSO ile entegre edilmiş gerçek kodlama da RSLFN'i iyileştirmek için mevcuttur [78-80]. RSLFN'in çıktı ağırlıkları iteratif olmayan öğrenme yoluyla elde edilebildiğinden, gerçek bir dizi olarak

kodlanan kalan parametreler bir birey olarak kabul edilir. Popülasyondaki i. birey X_i Eşitlik (20)'de belirtildiği gibidir:

$$X_i = (w_{11}, w_{12}, \dots, w_{1\tilde{N}}, w_{21}, w_{22}, \dots, w_{2\tilde{N}}, \dots, w_{n1}, w_{n2}, \dots, w_{n\tilde{N}}, b_1, b_2, \dots, b_{\tilde{N}}) \quad (20)$$

Burada n ve $\tilde{N}$ sırasıyla girdi düğümlerinin ve gizli düğümlerin sayısıdır. w_{jk}, j. girdi düğümünü k. gizli düğüme bağlayan ağırlıktır ve b_k, k. gizli düğümün önyargısıdır. X_i'nin bileşenleri, iterasyonlar sırasında [-1,1] aralığında sınırlandırılır. Sadece RSLFN'in girdi ağırlıklarının ve gizli önyargılarının ikili kodlama ve gerçek kodlama ile ifade edildiğine dikkat edin. Bu gerçek, popülasyondaki bireylerin boyutunu önemli ölçüde azaltır, bu da optimal parametreleri keşfetmek için nispeten küçük bir arama uzayına yol açar.

3.1.2. Değerlendirme mekanizması

Meta-sezgisel optimizasyon, sınırlı iterasyonlar sırasında ideale yakına ulaşabilir, bu da kısmen makul değerlendirme mekanizmasına, yani uygunluk fonksiyonu atamasına dayanır. Genellikle, daha küçük ağ çıktı hatasına sahip RSLFN genellikle daha iyi yaklaşım yeteneğine sahiptir. Bu nedenle, maliyet fonksiyonu olarak formüle edilen ağ çıktı hatası (bkz. Eşitlik (3)) değerlendirme kriteri olarak alınır. Ayrıca, Eşitlik (21) olarak doğrulama setindeki kök ortalama kare hatası (RMSE), bireyleri değerlendirmek için en yaygın kullanılan uygunluk fonksiyonudur.

$$f(T, H\beta) = \sqrt{\frac{\sum_{i=1}^{N_v} \left|\left|\sum_{j=1}^{\tilde{N}} \beta_j g(w_j \cdot x_i + b_j) - t_i\right|\right|_2^2}{N_v}} \quad (21)$$

Burada N_v doğrulama setinin sayısıdır.Tüm eğitim seti yerine doğrulama setinde değerlendirme yapmak, aşırı öğrenmeyi önlemek için etkili bir strateji olarak önerilmektedir [78,79]. Ancak, değerlendirme kriteri olarak tek başına doğrulama RMSE'sini kullanmak, RSLFN'in öğrenme yeteneğini ve genelleme performansını aynı anda artırmayabilir. Bartlett'ten [81], daha küçük ağırlıklara sahip sinir ağı daha iyi genelleme performansına sahip olma eğilimindedir. Bu nedenle, birçok çalışma optimizasyon sürecinde daha küçük β'ya sahip RSLFN'i seçmeye meyilliydi. [82]'de, büyük ağırlıklara sahip bireyleri cezalandırmak için uygunluk fonksiyonuna bir düzenlileştirme terimi $\left|\left|\beta\right|\right|$ eklenmiştir. Diğer bazı ilgili çalışmalar, DE'deki seçim operatörüne (Eşitlik (22)'de gösterildiği gibi) [78] veya PSO'daki parçacık güncelleme sürecine (Eşitlik (23) ve (24)'te gösterildiği gibi) [83,84] gömülebilecek karşılık gelen kısıtlı kriteri alternatif olarak ortaya koymuştur.

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & if\ f(X_{i,G}) - f(U_{i,G+1}) > \epsilon f(X_{i,G}) \\ U_{i,G+1} & if\ |f(X_{i,G}) - f(U_{i,G+1})| < \epsilon f(X_{i,G})\ ve\ ||\beta^{U_{i,G+1}}|| < ||\beta^{X_{i,G}}|| \\ X_{i,G} & diğer\ durumlar \end{cases} \quad (22)$$

$$P_{ib} = \begin{cases} X_i & if\ f(P_{ib}) - f(X_i) > \lambda f(P_{ib}) \\ X_i & if |f(P_{ib}) - f(X_i)| < \lambda f(P_{ib})\ ve\ ||\beta^{X_i}|| < ||\beta^{P_{ib}}|| \\ P_{ib} & diğer\ durumlar \end{cases} \quad (23)$$

$$P_g = \begin{cases} X_i & if\ f(P_g) - f(X_i) > \lambda f(P_g) \\ X_i & if |f(P_g) - f(X_i)| < \lambda f(P_g)\ ve\ ||\beta^{X_i}|| < ||\beta^{P_g}|| \\ P_g & diğer\ durumlar \end{cases} \quad (24)$$

Burada $f(\cdot)$ uygunluk fonksiyonudur (doğrulama hatası) ve ϵ, λ önceden ayarlanmış tolerans oranlarıdır. Eşitlik (22)-(24)'ten, iterasyonlar sırasında iki bireyden gelen uygunluk fonksiyonları arasındaki fark küçük olduğunda daha küçük $||\beta||$ ile sonuçlanan seçilir.

3.1.3. Girdi ağırlıkları ve gizli önyargıların optimizasyonu

Gerçek kodlamalı meta-sezgisel yöntemler, önceki çalışmalarda RSLFN ağırlık optimizasyonuna uygulanmıştır [80,85,86]. Popülasyon çaprazlama, mutasyon ve seçim operatörleri aracılığıyla evrilir ve en iyi girdi ağırlıkları ve gizli önyargı seti sonunda GA tarafından elde edilir. Ayrıca DE, GA'daki mutasyon ve çaprazlama operatörünü taklit ederek gerçek değerli ağırlık optimizasyonunu verimli bir şekilde ele alır [78,87-90]. Zhu ve ark. [78] ilk olarak gizli düğümleri optimize etmek için DE yöntemiyle evrimsel RSLFN önermiştir. Yöntem DE'yi değiştirmiş ve yeni seçim operatörü olarak Eşitlik (22)'yi benimsemiştir. Bazi ve ark. [89], hiperspektral görüntüleri sınıflandırmak için DE ile optimize edilmiş RSLFN kullanmıştır. Çalışma, DE'nin performansını artırmak için ek bir çaprazlama operatörü olarak ortogonal çaprazlamayı tanıtmış ve uygunluk fonksiyonu olarak çapraz doğrulama doğruluğunu benimsemiştir.

En popüler sürü zekası yöntemlerinden biri olan PSO, popülasyonu ideale yakın RSLFN ağırlık vektörlerine doğru yönlendirir [79,82-84,91-95]. Xu ve Shu [91] ilk olarak RSLFN'i eğitmek için hem iteratif olmayan öğrenmenin hem de PSO'nun avantajlarından yararlanmayı önermiştir. Pacifico ve Ludermir [83], RSLFN girdi ağırlıklarını ve gizli önyargılarını optimize etmek için PSO ve kümeleme analizini kullanmayı önermiştir. Bu yöntemde, her bireyin yalnızca kendi yakın komşuluğuna ait bazı üyeleri takip ettiği PSO popülasyon güncellemesi için yerel bir en iyi komşuluk şeması kullanılmıştır. Ling ve ark. [84], girdi ağırlıklarını ve gizli önyargıları optimize etmek için

RSLFN'in girdi-çıktı duyarlılık bilgisini kodlayan geliştirilmiş bir PSO geliştirmiştir. Ayrıca, yapay arı kolonisi [96], yapay balık sürüsü [97], grup araması [98] ve diğer sürü tabanlı yöntemlerin tümü bazı gerçekçi uygulamalarda RSLFN ağırlık optimizasyonu için mevcuttur [99-103].

Önceki çalışmalarda RSLFN'i optimize etmek için memetik algoritma [104], harmoni araması [105] vb. gibi birçok başka meta-sezgisel yaklaşım biçimi kullanılmıştır. Bu yöntemler, evrimsel algoritmalar ve sürü zekası yaklaşımları ile birlikte, optimal girdi ağırlıklarını ve gizli önyargıları aramaya yardımcı olan ve böylece RSLFN'in genelleme performansını önemli ölçüde artıran topluluğu oluşturmuştur.

3.2. Hem ağ mimarisinin hem de girdi ağırlıklarının optimizasyonu

SLFN'in öğrenme süreci RSLFN ile basitleştirilmiştir, ancak ağ topolojisi hala ampirik olarak önceden ayarlanmış ve sabittir. Optimal sayıda gizli düğüme sahip RSLFN genellikle önemli öğrenme yeteneğine ve genelleme performansına sahiptir. Eskiden, deneme yanılma yöntemi ağ mimarisi tasarımı için uygulanabilir bir yol olarak kabul edilirdi. Ancak, bu büyük ölçüde öznel deneyime bağlıdır. Bu nedenle, meta-sezgisel algoritmalar optimal topolojiyi otomatik olarak arayarak ağ mimarisi optimizasyonu araştırmalarının ilerlemesini teşvik eder [65].

Huang ve Lai [106], ampirik risk ve VC güveninden oluşan yapısal risk minimizasyonu (SRM) ile bir gizli düğüm optimizasyon yöntemi önermiştir. RSLFN nöronlarının sayısı $\widetilde{N}$ tek boyutlu bir parçacık olarak kodlanmış ve SRM fonksiyonunu optimize etmek için PSO uygulanmıştır [106]. Alencar ve ark. [107] GA aracılığıyla gizli nöronların optimal alt kümesini seçmiştir. $\widetilde{N}$ gizli düğümlü RSLFN ilk olarak sunulmuş ve GA'daki her kromozom $X_i = [x_1, x_2, \dots, x_{\widetilde{N}}]$ olarak modellenmiştir, burada $x_j \in 0{,}1$ aday alt kümeden j. düğümdür. Xue ve ark. [108,109] ayrıca gizli düğüm sayısını, girdi ağırlıklarını ve önyargıları aynı anda optimize etmek için değişken uzunluklu bir PSO araştırmıştır. Farklı ağ yapılandırmalarını ifade eden farklı uzunluktaki parçacıklar, bu çalışmada önerilen yeni bir parçacık güncelleme stratejisi ile ele alınabilir.

[110]'da, Fahlman ve Lebiere tarafından uygun mimariye sahip bir sinir ağı oluşturmak için bir kademeli-korelasyon (cascade-correlation) öğrenme modeli önerilmiştir. Geleneksel yapıcı yöntemin yüksek bir hesaplama maliyeti olduğundan, yeni bir tane eklendiğinde mevcut gizli nöronların ağırlıklarını donduran artımlı RSLFN (I-RSLFN) önerilmiştir [111]. Matematiksel olarak, yeni gizli düğüm ile çıktı nöronları arasındaki çıktı ağırlıkları $\beta_{\widetilde{N}_{new}}$ aşağıdaki prensibe göre hesaplanır.

$$\beta_{\tilde{N}_{new}} = \frac{E \cdot H_{\tilde{N}_{new}}^{T}}{H_{\tilde{N}_{new}} \cdot H_{\tilde{N}_{new}}^{T}} \tag{25}$$

Burada $H_{\tilde{N}_{new}}$ yeni gizli düğümün aktivasyonudur ve E yeni olan eklenmeden önceki artık hatadır. Uygulamada, sıfır yaklaşım hatası peşinde sonsuz düğüm eklemekten kaçınmak için normalde maksimum bir gizli düğüm sayısı veya bir hata eşiği verilir.

Pek çok çalışma I-RSLFN'i birden fazla açıdan geliştirmeyi önermiştir [112,113]. [112]'de, I-RSLFN için geliştirilmiş bir yöntem (EI-RSLFN olarak anılır) önerilmiştir. Seçilen düğüm mevcut ağa eklenmiş ve çıktı ağırlığı I-RSLFN'dekiyle aynı şekilde hesaplanmıştır. Ek olarak, rastgele ağırlıklı hatası en aza indirilmiş RSLFN (EM-RSLFN) [112], iteratif olmayan RSLFN'e rastgele gizli düğümleri tek tek veya değişen grup boyutlarıyla grup grup eklemek için önerilmiştir.

[114]'te, paralel kaos araması ile optimize edilen yeni düğüm, I-RSLFN'in her öğrenme adımında eklenmiştir. EM-RSLFN'i iyileştirmek için, [115,116]'da gizli düğüm sayısını, girdi ağırlıklarını ve gizli önyargıları optimize etmek için PSO kullanılmıştır. Optimizasyon sürecinde, yeni düğüm maksimum düğüm sayısı veya minimum artık hata elde edilene kadar mevcut RSLFN'e tek tek eklenmiştir. Yeni bir gizli düğüm için, çıktı ağırlıkları EM-RSLFN'in yaptığı gibi güncellenmiş ve daha düşük doğrulama RMSE'sine ve gizli çıktı matrisi koşul değerine sahip girdi ağırlıkları ve gizli önyargılar seti seçilmiştir. [117]'de, değişken uzunluklu artımlı RSLFN ve PSO'ya dayalı hibrit bir yaklaşım önerilmiştir.

4. Meta-sezgisel yaklaşımlara dayalı topluluk RSLFN

Topluluk (ensemble) öğrenimi, dikkate değer performansı nedeniyle ML topluluğunda büyük ilgi görmüştür [118,119]. Bagging [120], Boosting [121] ve Stacking [122] dahil olmak üzere en popüler topluluk öğrenimi yaklaşımları, bir temel öğrenici popülasyonunu sırayla veya paralel olarak eğitir ve ardından popülasyonu çeşitli stratejileri göz önünde bulundurarak birleştirir.

Temel öğrenicilerin ağırlıklarının belirlenmesi, daha iyi genelleme elde etmek için kritik bir teknik olarak görülmektedir, ancak bu hala açık bir sorudur. Topluluk budama, eğitilmiş tüm adayları birleştirmek yerine temel öğrenicilerin yalnızca bir alt kümesini seçer. Bunlar arasında, topluluğun çeşitliliği, topluluğun birkaç üyesinin bir girdi örüntüsü üzerinde aynı fikirde olmaması olarak tanımlanır [123,124,125,126]. Birçok literatür yukarıdaki problemleri optimizasyon görevleri olarak ele almakta ve meta-sezgisel algoritmaların bu optimizasyon problemlerinde yetkin olduğunu göstermektedir [123,127]. RSLFN topluluğu çoğu durumda tek RSLFN'den daha iyi performans gösterdiğinden [128,129], bazı çalışmalarda RSLFN

topluluğunu daha da iyileştirmek için meta-sezgisel algoritmalar kullanılmıştır [130,131]. Bu bölüm, meta-sezgisel ve RSLFN topluluğunun kombinasyonuna ilişkin katkıları, özellikle topluluk üyelerinin ve topluluk ağırlıklarının optimizasyonuna odaklanarak kapsamlı bir şekilde incelemeyi amaçlamaktadır.

4.1. Topluluk ağırlıkları optimizasyonu

Gerçek kodlamalı meta-sezgisel algoritmalar, daha iyi tahmin performansı için topluluk ağırlıklarını optimize etmek üzere kullanılabilir. RSLFN topluluk ağırlıklarının optimizasyonu için, meta-sezgisel algoritmadaki i. birey X_i, temel RSLFN'lere verilen gerçek değerli ağırlıklar w_j'den oluşur. Başlatılan ağırlıklar öncelikle aşağıdaki denklemi sağlamalıdır:

$$\sum_{j=1}^{N_E} \omega_j = 1{,}0 < \omega_j < 1 \tag{26}$$

Burada ω_j, j. temel RSLFN'in ağırlığıdır ve N_E topluluğun boyutudur. Ayrıca, topluluk ağırlıklarının kombinasyonunu değerlendirmek için uygunluk fonksiyonu olarak sınıflandırma doğruluğu veya RMSE kabul edilir.

DE, RSLFN topluluğuna ağırlık kombinasyonunu aramak için başarıyla kullanılmıştır [132,133]. [132]'de, ağırlıklar gerçek kodlama şeması ile DE algoritması tarafından optimize edilmiş ve ardından yeni bir topluluk oluşturmak için daha büyük ağırlıklara sahip temel RSLFN'ler seçilmiştir. Gerçek kodlu bireylere sahip DE, [133]'te ağırlıklı RSLFN topluluğu için de kullanılmıştır. Dengesiz verilerle başa çıkmak amacıyla, [133]'teki yöntem, RSLFN topluluğunun sınıflandırma performansını değerlendirmek için uygunluk fonksiyonu olarak duyarlılık ve özgüllüğün çarpımının karekökünü belirlemiştir. PSO tabanlı yönteme dönersek, Liu ve ark. [134], çevrimiçi sıralı RSLFN (OS-RSLFN) topluluğunun ağırlıklarını optimize etmek için PSO kullanmayı önermiştir. Ağırlığı önceden ayarlanmış bir eşikten büyük olan temel OS-RSLFN, PSO işleminden sonra seçilmiştir. Ling ve ark. [135], topluluk ağırlıklarını optimize etmek için çekici ve itici PSO (ARPSO) [136] kullanmayı önermiştir.

4.2. Topluluk budama optimizasyonu

RSLFN topluluk budaması, genelleme performansını iyileştirmek için bir topluluk modelindeki üyelerin azaltılmasını ele almaktır. Kullanılabilir bir seçici topluluk oluşturmak bir optimizasyon problemi olarak kabul edilir. Meta-sezgisel ile RSLFN topluluk budaması üzerine araştırma, gelecek vadeden bir alandır çünkü meta-sezgisel, uygun bir seçici topluluğu doğrudan veya dolaylı olarak arayabilir.

4.2.1. Topluluktaki temel RSLFN için kodlama şeması

En iyi seçici RSLFN topluluğunu bulmak için, meta-sezgisel kodlama şeması en önemli hususdur. Önceki çalışmalar, RSLFN topluluk budamasında gerçek kodlama yerine ikili kodlamayı tercih etmiştir. GA'da, başlangıç havuzundaki her RSLFN bir ikili diziye kodlanır ve seçici topluluklar ikili kromozomlarla ifade edilir [119].

Örneğin, dört RSLFN'li bir havuz varsayalım, tüm topluluk aşağıdaki gibi 2 bitlik dizi ile tanımlanabilir: $RSLFN_1 = 00, RSLFN_2 = 01, RSLFN_3 = 10, RSLFN_4 = 11$. Topluluk boyutu 2'ye eşit olduğunda, olası seçici toplulukları temsil eden kromozomlar aşağıdaki gibi gösterilir: $ch_1 = 0001, ch_2 = 0010, ch_3 = 0011, ch_4 = 0110, ch_5 = 0111, ch_6 = 1011$.

Yukarıdaki tartışmadan, ikili kodlamalı GA'nın yalnızca sabit bir boyuta sahip optimal seçici topluluğu arayabileceği açıktır. Bu nedenle, Wang ve Alhamdoosh [119], boyut kontrol stratejisi ile en iyi topluluğu aramayı önermiştir. Topluluk boyutu K, GA yakınsadığında artırılmıştır. Sağlanan bir K ile kromozomun uzunluğuna karar verilmiş ve K RSLFN'li en iyi seçici topluluk GA aracılığıyla elde edilmiştir. Topluluk boyutu artışı, topluluk genelleme hatası arttığında veya maksimum boyuta ulaşıldığında nihayet durmuştur.

GA'ya ek olarak, ikili kodlama bu durumda PSO tabanlı yöntemlerle entegre edilmiştir. Kodlama uzunluğu, başlangıç topluluk havuzunun boyutuyla belirlenir. Özellikle, 1 veya 0 ile ifade edilen parçacığın i. bileşeni, i. RSLFN'in seçildiği veya seçilmediği anlamına gelir. PSO'nun nihai çözümü, budanmış toplulukta hangilerinin seçildiğini açıklayan bir dizidir. Han ve ark. [137-139], ikili kodlama ile topluluk budaması için ARPSO kullanmayı önermiştir. Değiştirilmiş ARPSO'da, parçacık güncelleme kuralı geleneksel ARPSO'yu izlemiş ve parçacığın j. bileşeninin değeri her yinelemede değerlendirmeden önce 1 veya 0 olarak yuvarlanmıştır.

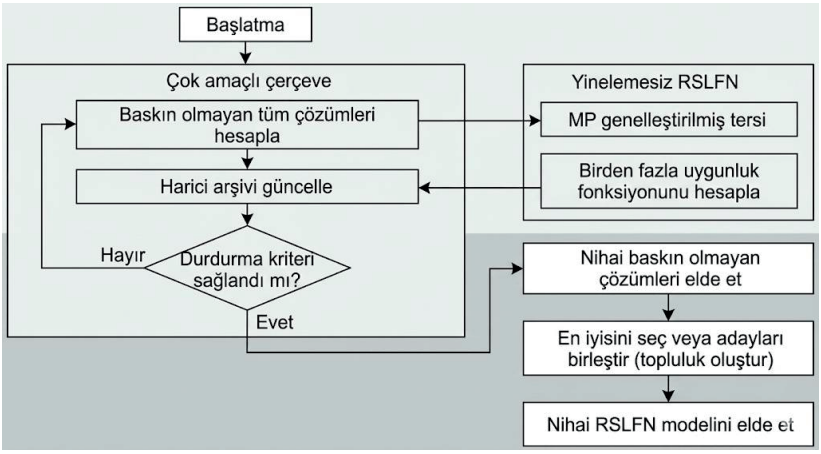
4.2.2. RSLFN topluluk budaması için değerlendirme kriteri

Mevcut çalışmalar genellikle nesiller boyunca budanmış RSLFN topluluğunu değerlendirmek için uygunluk fonksiyonu olarak topluluk doğruluğunu benimser [132,135,140]. DE ile birleştirilmiş uygunluk paylaşımı (fitness sharing), [130]'da standart evrimsel algoritmalardan daha çeşitli sonuçlar üretmek için kullanılmıştır. Birbirine benzeyen bireyler uygunluklarını paylaşmaları istenerek cezalandırılırken, izole bireyler elde ettikleri tüm uygunluk değerini korumuştur. [119]'daki yöntem, temel RSLFN ağırlıkları ve genelleme hatası ile ilgili olan çeşitliliği GA'nın uygunluk fonksiyonu olarak bile görmüştür. [139]'da, geliştirilmiş genelleme performansına sahip kompakt bir topluluk elde etmek için çeşitlilik güdümlü bir yöntem

geliştirilmiştir. İki RSLFN arasındaki mesafeyi ölçen çeşitlilik aşağıdaki gibi tanımlanmıştır:

$$div = \frac{2}{N\times(N-1)}\sum_{k=1}^{N-1}\sum_{l=k+1}^{N}(\,||\omega_k - \omega_l||_2^2 + ||b_k - b_l||_2^2 + ||\beta_k - \beta_l||_2^2 + ||Y_k - Y_l||_2^2) \quad (27)$$

Burada ω_k ve ω_l, i. parçacıkta k. ve l. seçilen RSLFN'in girdi ağırlık matrisleridir, b_k ve b_l önyargı vektörleridir, β_k ve β_l çıktı ağırlıklarıdır, Y_k ve Y_l çıktı vektörleridir. Daha sonra çeşitlilik bilgisi, parçacıkları en iyiye yönlendirmek için optimizasyon sürecine kodlanmıştır.



Şekil 3: Pareto tabanlı çok amaçlı optimizasyona dayalı RSLFN'in genel çerçevesi

5. Çok amaçlı meta-sezgisel tabanlı RSLFN

Tek amaçlı optimizasyon yönteminin tek bir simülasyon çalışmasında benzersiz bir çözümü varken, çok amaçlı optimizasyon sonunda Pareto cephesine yaklaşan çok sayıda Pareto optimal çözümü sunar [141,142]. Çok amaçlı meta-sezgisel yaklaşımlar, birden fazla rakip hedefi paralel olarak ele almak için esasen meta-sezgiseli çok amaçlı problemle birleştirir. Bu hedefler, ön bilgi veya korelasyon bilgisi olmadan eşdeğer olarak ele alınmalıdır [143,144]. Yaklaşımlar sonunda, en iyisinin karar verici tarafından seçildiği çeşitli baskın olmayan çözümler kümesi üretir.

Yalnızca yaklaşım hatasını en aza indirmek RSLFN performansını iyileştirmek için yeterli olmadığından, aşağıdaki hedefler çok amaçlı meta-sezgisel yaklaşımlar için gerekli hale gelir. Her şeyden önce, denetimli öğrenme sorunu için, maliyet fonksiyonu (Eşitlik (3)), ağ yaklaşım yeteneğini garanti eden birincil hedeftir. İkinci olarak, genelleme performansını iyileştirmek RSLFN'i optimize etmek için önemlidir. Bilindiği gibi, çıktı

ağırlıklarını en aza indirmek ağ karmaşıklığını basitleştirmeye yol açar [81]. Bu nedenle, RSLFN'i optimize etmek için çok amaçlı bir problem şu şekilde modellenir:

$$\min {}_nF = (||H\beta - T||_2, ||\beta||_2)^T \tag{28}$$

Ayrıca, seyrek seviye de RSLFN'in genelleme performansı için önemlidir. Seyrek düzenlileştirme, gizli nöronların çoğunun sıfıra eşit veya sıfıra yakın olmasına neden olur, bu da biyolojik nöronların mekanizmasına çok daha yakındır [145]. Seyrek gizli temsil elde etmek için l_1 düzenlileştirmesini (bkz. Eşitlik (29)) veya Kullback-Leibler sapmasını (bkz. Eşitlik (30) ve (31)) en aza indirmek tipik olarak kullanılır [146].

$$\min ||H||_1 \tag{29}$$

$$min \sum_{j=1}^{\tilde{N}} K\, L(\rho||\widehat{\rho_j}) \tag{30}$$

$$KL(\rho||\hat{\rho}_j) = \rho\ log\frac{\rho}{\hat{\rho}_j} + (1-\rho) log\frac{1-\rho}{1-\hat{\rho}_j} \tag{31}$$

Burada ρ gizli aktivasyonun hedef değeridir, $\widehat{\rho_j} = \frac{1}{\tilde{N}}\sum_j^{\tilde{N}} g\left(w_j \cdot x_i + b_j\right)$ gerçek aktivasyondur.

Özetle, RSLFN'i geliştirmek için genel bir çok amaçlı optimizasyon problemi şu şekilde sunulur:

$$\min {}_3F = (f_{err}, f_{add})^T \tag{32}$$

Burada f_{err} ağ çıktı hatası (örneğin, doğrulama hatası) olarak kabul edilir. Ağırlıkların normu, gizli aktivasyon değerleri, ağ yapısı veya bunlardan bazılarının bir kombinasyonu ek bir hedef f_{add} olarak düşünülebilir [55].

RSLFN'i geliştirmek için çok amaçlı meta-sezgisel algoritmaların iki avantajı vardır. (1) Çoklu uygunluk fonksiyonları değerlendirmesi. Çözümler birden fazla uygunluk fonksiyonu ile değerlendirilir ve Pareto cephesine yönlendirilir. Bu durum, çok amaçlı yaklaşımların RSLFN'i Eşitlik (32)'de gösterildiği gibi birden fazla kritere tabi olarak optimize etmesini sağlar. (2) Seçkincilik (Elitism) veya harici depo [147,148].

Baskın olmayan sıralama genetik algoritması sürüm II (NSGA-II), en çok atıf yapılan Pareto tabanlı çok amaçlı optimizasyon algoritmasıdır [149]. NSGA-II'de Deb ve ark., hızlı bir baskın olmayan sıralama prosedürü ve seçkinci koruma stratejisi kullanan bir baskın olmayan sıralama GA yaklaşımı önermiştir. NSGA-II, çeşitli çalışmalarda geliştirilmiş RSLFN elde etmek için uygulanmıştır [150-152]. Cai ve ark. [150], NSGA-II kullanarak RSLFN ve RSLFN otokodlayıcısının optimal girdi ağırlıklarını ve önyargılarını aramayı önermiştir. Sınıflandırma görevleri için RSLFN, doğrulama setindeki

sınıflandırma hatasını (f_{err} olarak düşünün) ve gizli katmanın seyrekliğini (f_{add}, bkz. Eşitlik (30)) içeren iki çelişkili hedefle entegre edilmiş NSGA-II ile optimize edilmiştir. Özellik çıkarma görevi için RSLFN otokodlayıcısının optimizasyonu, doğrulama RMSE (f_{err}, bkz. Eşitlik (21)) ve seyrekliği (f_{add}) içeren çift hedeflere tabi tutulmuştur. Echanobe ve ark. da [151] girdi özelliklerinin boyutunu (f_{err}), RSLFN yapısını (f_{add}) ve sınıflandırma hatasını (f_{err}) en aza indirmek için NSGA-II kullanmayı ve ardından modeli çevrimiçi hiperspektral görüntü sınıflandırması için kullanmayı önermiştir. Du ve ark. [542] NSGA-II'yi geliştirmiş ve RSLFN'i üç hedefe tabi olarak optimize etmek için baskın olmayan sıralama uyarlamalı diferansiyel gelişim önermiştir: ağ çıktı hatası (f_{err}), çıktı ağırlıklarının l_2 normu (f_{err}, bkz. Eşitlik (28)) ve gizli düğüm sayısı (f_{add}).

Benzer şekilde, [153]'te önerilen çok amaçlı parçacık sürü optimizasyonu (MOPSO), RSLFN'i iyileştirmek için arama stratejisi olarak seçilmiştir. Mao ve ark. [154] ayrıca çok amaçlı kapsamlı öğrenme PSO (MOCLPSO) kullanmıştır [155]. Gizli düğümler RSLFN'e birer birer eklenmiş ve MOCLPSO, her adımda "leave-one-out" hata sınırını (f_{err}) [156] ve çıktı ağırlıklarının normunu (f_{add}) en aza indirerek optimal girdi ağırlıklarını seçmek için kullanılmıştır. Ek olarak, mikro genetik [157] ve membran sistemlerine [158] dayalı çok amaçlı optimizasyon da RSLFN'in optimal parametrelerini keşfetmek için sunulmuştur.

6. Zorluklar ve gelecek kapsamları

Hem RSLFN hem de meta-sezgisel, biyo-esinli içgörülerdir. RSLFN teorisi insan beyni öğrenme mekanizmasını takip eder ve rastgele nöronları biyolojik sistemdeki olası yerel düzensiz yapıyı simüle eder. Meta-sezgisel, doğal fenomenleri taklit ederek dışbükey olmayan ve NP-zor problemleri çözebilir, bu da onu diğer optimizasyon algoritmalarından bazılarına üstün kılar. Meta-sezgisel algoritma kullanmak, daha iyi yaklaşım yeteneğine ve genelleme performansına sahip geliştirilmiş bir RSLFN/RSLFN topluluğuna yol açacaktır.

Gelecekteki araştırmalarla ilgili olarak, eğilim aşağıdaki yönlere odaklanmaya doğrudur. İlk olarak, çok çekici bir yön hesaplama karmaşıklığını azaltmaktır. Meta-sezgisel yöntemle RSLFN genellikle hesaplama açısından yoğun olmaktan muzdariptir. Çok sayıda değerlendirme genellikle hesaplama maliyetinin çoğunu alır. Hesaplama maliyetini azaltmak için, vekil (surrogate) model [159,160] gibi hesaplama açısından verimli tekniklerin dikkate alınabileceği görülmektedir.

İkinci olarak, RSLFN için çok amaçlı optimizasyon çerçevesinin daha fazla araştırılması beklenmektedir. Yaklaşım hatasını en aza indirmek ve genelleme performansını artırmak ana hedefler olarak sınıflandırılır. Mevcut çok amaçlı yöntemleri iyileştirmek [161], rasyonel Pareto baskın mekanizması

tasarlamak [162] ve çok hedefli algoritmaları kullanmak [163] yukarıdaki açık sorunları çözmeye yardımcı olabilir.

Üçüncü olarak, evrimsel sinir ağları, özellikle evrimsel derin sinir ağları, yavaş yavaş araştırma odağı haline gelmiştir [164,165]. RSLFN derin ağa genişletildiğinden [166,167], çok katmanlı rastgele sinir ağını evrimleştirmek bir araştırma hattını teşvik edecektir. Meta-sezgisel ile entegre edilmiş hiyerarşik iteratif olmayan öğrenme modelinin etkileyici bir performans elde edeceğine inanılmaktadır. Dahası, çok katmanlı rastgele sinir ağını evrimleştirmek, nesne algılama, görüntü sınıflandırma ve özellikle pekiştirmeli öğrenme ile karmaşık uygulamalar [168,169] için çok uygun görünmektedir.

Son olarak, sınıf dengesizliği problemi üzerine araştırma da değerlidir [170,171]. Sınıf dengesizliği, bir veri kümesi diğer sınıflardan çok daha fazla örneğe sahip olan ana sınıflar tarafından domine edildiğinde ortaya çıkar. Problem, sınıflandırıcının yapımını ve doğruluğunu büyük ölçüde etkiler. Sınıf dengesizliği veri kümesiyle, örneğin mikrodizi verileriyle [172] başa çıkmak için hibrit RSLFN kullanmanın anlamlı ve zorlu olduğu açıktır.

Referanslar

[1] D.S. Huang, Systematic Theory of Neural Networks for Pattern Recognition, Publishing House of Electronic Industry of China, 1996.
[2] Y. Lecun, Y. Bengio, G. Hinton, Deep learning. Nature 521 (7553) (2015) 436-444.
[3] Y.W. Lu. N. Sundararajan, P. Saratchandran, A sequential learning scheme for function approximation using minimal radial basis function neural networks, Neural Comput. 9 (2) (1997) 461-478.
[4] S. Ferrari, R.F. Stengel, Smooth function approximation using neural networks, IEEE Trans. Neural Netw. 16 (1) (2005) 24-38.
[5] D.S. Huang, The Study of Data Mining Methods for Gene Expression Profiles, Science Press of China, 1996.
[6] S.C. Jiang. K.S. Chin, L. Wang, G. Qu, K.L. Tsui, Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department, Expert Syst. Appl. 82 (C) (2017) 216-230.
[7] D.S. Huang, Radial basis probabilistic neural networks: model and application, Int. J. Pattern Recognit. Artif. Intell. 13 (7) (1999) 1083-1101.
[8] L Shang, D.S. Huang, J.X. Du, C.H. Zheng, Palmprint recognition using FastICA algorithm and radial basis probabilistic neural network, Neurocomputing 69 (13) (2006) 1782-1786.
[9] Z.Q, Zhao, D.S. Huang, W. Jia, Palmprint recognition with 2DPCA+PCA based on modular neural networks, Neurocomputing 71 (1) (2007) 448-454.
[10] X.F. Wang, D.S. Huang, J.X. Du, H. Xu, L. Heutte, Classification of plant leaf images with complicated background. Appl. Math. Comput. 205 (2) (2008) 916-926.
[11] H.L. Luo, Y. Yang. B. Tong. F.C. Wu. B. Fan, Traffic sign recognition using a multi-task convolutional neural network, IEEE Trans. Intell. Transp. Syst. PP (99) (2017) 1-12.
[12] D. Silver, J. Schrittwieser, K. Simonyan, L Antonoglou, A. Huang. A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Mastering the game of Go without human knowledge, Nature 550 (7676) (2017) 354-359.
[13] D.S. Kermany, M. Goldbaum, W.J. Cai, C.C.S. Valentim, H.Y. Liang, S.L. Baxter, A. Mckeown, G. Yang, X.K. Wu, F.B. Yan, Identifying medical diagnoses and treatable diseases by image-based deep learning, Cell 172 (5) (2018) 1122-1131.
[14] DS. Huang, S.D. Ma, Linear and nonlinear feedforward neural network classifiers: a comprehensive understanding. J. Intell. Syst. 9 (1) (1999) 1-38.
[15] CH. Zheng, D.S. Huang, ZL Sun, M.R. Lyu, T.M. Lok, Nonnegative independent component analysis based on minimizing mutual information technique, Neurocomputing 69 (7-9) (2006) 878-883.
[16] C.H. Zheng, D.S. Huang, K. Li, G. Irwin, Z.L. Sun, MISEP method for postnonlinear blind source separation., Neural Comput. 19 (9) (2007) 2557-2578.
[17] S. Ganjefar, M. Tofighi, Single-hidden-layer fuzzy recurrent wavelet neural network: applications to function approximation and system identification. Inf. Sci. 294 (2015) 269-285. [18] D.S. Huang, A constructive approach for finding

arbitrary roots of polynomials by neural networks, IEEE Trans. Neural Netw. 15 (2) (2004) 477-491.
[19] D.S. Huang, 2. Chi, Finding roots of arbitrary high order polynomials based on neural network recursive partitioning method, Sci. Chin. Ser. F Inf. Sci. 47 (2) (2004) 232-245.
[20] DS. Huang, H.H.S. Ip, Z. Chi, A neural root finder of polynomials based on root moments, Neural Comput. 16 (8) (2004) 1721-1762.
[21] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Parallel Distrib. Process. Explor. Microstruct. Cognit. 323 (6088) (1986) 399-421.
[22] P. Patrick van der Smagt, Minimization methods for training feed forward neural networks, Neural Netw. 7 (1) (1994) 1-11.
[23] R. Battiti, First and second order methods for learning: between steepest descent and Newton's method, Neural Comput. 4 (2) (1992) 141-166.
[24] R. Johnson, T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, in: Proceedings of the International Conference on Neural Information Processing Systems, 1, 2013, pp. 315-323.
[25] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations, 2014, pp. 1-13.
[26] B.G. Hu, H.B. Qu, Y. Wang, S.H. Yang, A generalized-constraint neural network model: associating partially known relationships for nonlinear regressions, Inf. Sci. 179 (12) (2009) 1929-1943.
[27] S.Y. Jeong, S.Y. Lee. Adaptive learning algorithms to incorporate additional functional constraints into neural networks, Neurocomputing 35 (1) (2000) 73-90.
[28] DS. Huang, H.H.S. Ip. Z. Chi, H.S, Wong. Dilation method for finding close roots of polynomials based on constrained learning neural networks, Phys Lett. A 309 (56) (2003) 443-451.
[29] DS. Huang, H.H.S. Ip, K.C. Law, 2. Chi, Zeroing polynomials using modified constrained neural network approach, IEEE Trans. Neural Netw. 16 (3) (2005) 721-732.
[30] DS. Huang, W.B. Zhao, Determining the centers of radial basis probabilistic neural networks by recursive orthogonal least square algorithms, Appl. Math. Comput. 162 (1) (2005) 461-473.
[31] JX. Du, D.S. Huang, G.J. Zhang, Z.F. Wang, A novel full structure optimization algorithm for radial basis probabilistic neural networks, Neurocomputing 70 (1) (2006) 592-596.
[32] D.S. Huang. J.X. Mi. A new constrained independent component analysis method, IEEE Trans. Neural Netw. 18 (5) (2007) 1532-1535.
[33] W.H. Joerding, J.L. Meador, Encoding a priori information in feedforward networks, Neural Netw. 4 (6) (1991) 847-856.
[34] J. Murata, K. Hirasawa, A new learning method using prior information of neural networks, Sci. Chin. Ser. F Inf. Sci. 47 (6) (2004) 793-814.
[35] F. Han, D.S. Huang, Improved extreme learning machine for function approximation by encoding a priori information, Neurocomputing 69 (1618) (2006) 2369-2373.
[36] F. Han, D.S. Huang. A new constrained learning algorithm for function approximation by encoding a priori information into feedforward neural networks, Neural Comput. Appl. 17 (5-6) (2008) 433-439.

[37] E. Han, Q.H. Ling, D.S. Huang, Modified constrained learning algorithms incorporating additional functional constraints into neural networks, Inf. Sci. 178 (3) (2008) 907-919.

Elbette, gönderdiğiniz referans listesini her biri yeni bir satırda başlayacak şekilde düzenledim:

[38] F. Han, Q.H. Ling, D.S. Huang, An improved approximation approach incorporating particle swarm optimization and a priori information into neural networks, Neural Comput. Appl. 19 (2) (2010) 255-261.

[39] Y.H. Pao, S. Phillips, D.J. Sobajic, Neural-net computing and the intelligent control of systems, Int. J. Control 56 (2) (1992) 263-289.

[40] L. Zhang, P.N. Suganthan, A comprehensive evaluation of random vector functional link networks, Inf. Sci. 367-368 (2016) 1094-1105.

[41] Y.H. Pao, G.H. Park. D.J. Sobajic, Learning and generalization characteristics of the random vector functional-link net, Neurocomputing 364-365 (2) (1994) 163-180.

[42] D. Husmeier, Random Vector Functional Link (RVFL) Networks, Springer, London, 1999.

[43] B. Igelnik, Y.H. Pao, Stochastic choice of basis functions in adaptive function approximation and the functional-link net, IEEE Trans. Neural Netw, 6 (6) (1995) 1320-1329.

[44] Y.H. Pao, S.M. Phillips, The functional link net and learning optimal control, Neurocomputing 9 (2) (1995) 149-164.

[45] L Zhang, P.N. Suganthan, A survey of randomized algorithms for training neural networks, Inf. Sci. 6 (2016) 146-155.

[46] W.F. Schmidt, M.A. Kraaijveld, R.P.W. Duin, Feedforward neural networks with random weights, in: Proceedings of the International Conference on Pattern Recognition, 2002, pp. 1-4.

[47] Y. Ren, P.N. Suganthan, N. Srikanth, G. Amaratunga, Random vector functional link network for short-term electricity load demand forecasting, Inf. Sci. 367 (C) (2016) 1078-1093.

[48] A.M. Reznik, Non-iterative learning for neural networks, in: Proceedings of the International Joint Conference on Neural Networks, 2, 1999, pp. 1374-1379.

[49] W.F. Schmidt, M.A. Kraaijveld, R.P.W. Duin, A non-iterative method for training feedforward networks, in: Proceedings of the International Joint Conference on Neural Networks, 2, 2002, pp. 19-24.

[50] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: Proceedings of the IEEE International Joint Conference on Neural Networks, 2004, pp. 985-990.

[51] W.B. Zheng, Y.T. Qian, H.J. Lu, Text categorization based on regularization extreme learning machine, Neural Comput. Appl. 22 (3-4) (2013) 447-456.

[52] M.X. Luo, K. Zhang. A hybrid approach combining extreme learning machine and sparse representation for image classification, Eng. Appl. Artif. Intell. 27 (C) (2014) 228-235.

[53] J.M. Xu, W.Q. Zhang, J. Liu, S.H. Xia, Regularized minimum class variance extreme learning machine for language recognition, EURASIP J. Audio Speech Music Process. 2015 (1) (2015) 1-10.

[54] J. Zhang, D.S. Huang, T.M. Lok, M.R. Lyu, A novel adaptive sequential niche technique for multimodal function optimization, Neurocomputing 69 (1618) (2006) 2396-2401.

[55] A. Gogna, A. Tayal, Metaheuristics: review and application, J. Exp. Theoret. Artif. Intell. 25 (4) (2013) 503-526.
[56] K. Socha, C. Blum. An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training, Neural Comput. Appl. 16 (3) (2007) 235-247.
[57] W.B. Zhao, D.S. Huang, J.Y. Du, L.M. Wang, Genetic optimization of radial basis probabilistic neural networks, Int. J. Pattern Recognit, Artif. Intell. 18 (08) (2008) 1473-1499.
[58] D.S. Huang, J.X. Du, A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks, IEEE Trans. Neural Netw. 19 (12) (2008) 2099-2115.
[59] X Yao, Y. Liu, A new evolutionary system for evolving artificial neural networks, IEEE Trans. Neural Netw. 8 (3) (2002) 694-713.
[60] K.D. Stanley, R. Miikkulainen, Evolving neural networks through augmenting topologies, Evolut. Comput. 10 (2) (2002) 99-127.
[61] Z.Q. Zhao, D.S. Huang. A mended hybrid learning algorithm for radial basis function neural networks to improve generalization capability, Appl. Math. Model. 31 (7) (2007) 1271-1281.
[62] J.X. Du, D.S. Huang, X.F. Wang, X. Gu, Shape recognition based on neural networks trained by differential evolution algorithm, Neurocomputing 70 (46) (2007) 896-903.
[63] X. Yao, Evolving artificial neural networks, Proc. IEEE 87 (9) (1999) 1423-1447.
[64] S.F. Ding. H. Li, C.Y. Su, J.Z. Yu, F.X. Jin, Evolutionary artificial neural networks: a review, Artif. Intell. Rev. 39 (3) (2013) 251-260.
[65] V.K. Ojha, A. Abraham, V. Snášel, Metaheuristic design of feedforward neural networks: a review of two decades of research. Eng. Appl. Artif. Intell. 60 (2017) 97-116
[66] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, University of Michigan Press, 1975.
[67] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evolut. Comput, 15 (1) (2011) 4-31.
[68] M.R. Bonyadi, Z. Michalewicz, Particle swarm optimization for single objective continuous space problems a review., Evolut. Comput. 25 (1) (2017) 1-54.
[69] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning. Mach Learn. 3 (2) (1988) 95-99.
[70] ZL Sun, D.S. Huang, CH. Zheng, L. Shang. Optimal selection of time lags for TDSEP based on genetic algorithm, Neurocomputing 69 (79) (2006) 884-887.
[71] R. Storn, K. Price, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341-359.
[72] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, 4, 1995, pp. 1942-1948.
[73] J. Kennedy, R. Mendes, Population structure and particle swarm performance, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2, 2002. pp. 1671-1676.
[74] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evolut. Comput. 8 (2) (2014) 173-195.

[75] D. Whitley. The GENTTOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best, in: Proceedings of the International. Conference on Genetic Algorithms, 1989, pp. 116-121.
[76] S.F. Ding, C.Y. Su, J.Z. Yu, An optimizing BP neural network algorithm based on genetic algorithm, Artif. Intell. Rev. 36 (2) (2011) 153-162.
[77] F. Petroski Such, V. Madhavan, E. Conti, J. Lehman, K. Stanley, J. Clune, Deep neuroevolution: genetic algorithms are a competitive alternative for training Deep Neural Netw. Reinfor. Learn., 2017. arXiv: 1712.06567.
[78] Q.Y. Zhu, A.K. Qin. P.N. Suganthan, G.B. Huang, Evolutionary extreme learning machine, Pattern Recognit. 38 (10) (2005) 1759-1763.
[79] F. Han, H.F. Yao, Q.H. Ling. An improved evolutionary extreme learning machine based on particle swarm optimization, Neurocomputing 116 (2013) 87-93.
[80] X.W. Xue, M. Yao, Z.H. Wu, J.H. Yang, Genetic ensemble of extreme learning machine, Neurocomputing 129 (10) (2014) 175-184.
[81] P.L. Bartlett, The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network, IEEE Trans, Inf. Theory 44 (2) (1998) 525-536.
[82] Z.P. Yang, X.X. Wen, Z.Q. Wang, QPSO-ELM: an evolutionary extreme learning machine based on quantum-behaved particle swarm optimization, in: Proceedings of the International Conference on Advanced Computational Intelligence, 2015, pp. 69-72.
[83] L.D.S. Pacifico, T.B. Ludermir. Evolutionary extreme learning machine based on particle swarm optimization and clustering strategies, in: Proceedings of the International Joint Conference on Neural Networks, 2014, pp. 1-6.
[84] Q.H. Ling, Y.Q. Song, F. Han, H. Lu, An improved evolutionary random neural networks based on particle swarm optimization and input-to-output sensitivity, in: Proceedings of the International Conference on Intelligent Computing, 2017, pp. 121-127.
[85] S. Suresh, S. Saraswathi, N. Sundararajan, Performance enhancement of extreme learning machine for multi-category sparse data classification problems, Eng. Appl. Artif. Intell. 23 (7) (2010) 1149-1157.
[86] H.M. Yang, J. Yi, J.H. Zhao, ZY. Dong, Extreme learning machine based genetic algorithm and its application in power system economic dispatch. Neurocomputing 102 (2) (2013) 154-162.
[87] Y.P. Qu, C.J. Shang, W. Wu, Q. Shen, Evolutionary fuzzy extreme learning machine for mammographic risk analysis., Int. J. Fuzzy Syst. 13 (4) (2011) 282-291.
[88] K. Li, R. Wang, S. Kwong. J.J. Cao, Evolving extreme learning machine paradigm with adaptive operator selection and parameter control, Int. J. Uncertain. Fuzz. Knowl. Based Syst. 21 (supp02) (2013) 143-154.
[89] Y. Bazi, N. Alajlan, F. Melgani, H. Alhichri, S. Malek, R.R. Yager, Differential evolution extreme learning machine for the classification of hyperspectral images, IEEE Geosci. Remote Sens. Lett. 11 (6) (2014) 1066-1070.
[90] T. Matias, F. Souza, C.H. Antunes, Learning of a single-hidden layer feedforward neural network using an optimized extreme learning machine, Neurocomputing 129 (129) (2014) 428-436.

[91] Y. Xu, Y. Shu, Evolutionary extreme learning machine based on particle swarm optimization, in: Proceedings of the International Conference on Advances in Neural Networks, 2006, pp. 644-652.
[92] S. Saraswathi, S. Sundaram, N. Sundararajan, M. Zimmermann, M. Nilsen-hamilton, ICGA-PSO-ELM approach for accurate multiclass cancer classification resulting in reduced gene sets in which genes encoding secreted proteins are highly represented, IEEE/ACM Trans. Comput. Biol. Bioinf. 8 (2) (2011) 452-463.
[93] H.J. Lu, B.J. Du, J.Y. Liu, H.X. Xia, W.K. Yeap, A kernel extreme learning machine algorithm based on improved particle swam optimization, Memetic Comput. 9 (2) (2016) 1-8.
[94] N.Y. Zeng. H. Zhang, W.B. Liu, J.L. Liang, F.E. Alsaadi, A switching delayed PSO optimized extreme learning machine for short-term load forecasting, Neurocomputing 240 (2017) 175-182.
[95] E.M.N. Figueiredo, T.B. Ludermir, E.M.N. Figueiredo, T.B. Ludermir, Investigating the use of alternative topologies on performance of the PSO-ELM. Neurocomputing 127 (3) (2014) 4-12.
[96] D. Karaboga, B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim. 39 (3) (2007) 459-471.
[97] C.R. Wang, CL. Zhou, J.W. Ma, An improved artificial fish-swarm algorithm and its application in feed-forward neural networks, in: Proceedings of the International Conference on Machine Learning and Cybernetics, 5, 2005, pp. 2890-2894.
[98] S. He. Q.H. Wu, J.R. Saunders, Group search optimizer: an optimization algorithm inspired by animal searching behavior, IEEE Trans. Evolut. Comput. 13 (5) (2009) 973-990.
[99] X.F. Tang. L. Chen, Artificial bee colony optimization-based weighted extreme learning machine for imbalanced data learning. Cluster Comput. (2018) 1-16.
[100] S. Li, P. Wang, L. Goel, Short-term load forecasting by wavelet transform and evolutionary extreme learning machine, Electr. Power Syst. Res. 122 (2015) 96-103.
[101] J.FL.D. Oliveira, T.B. Ludermir. A modified artificial fish swarm algorithm for the optimization of extreme learning machines, in: Proceedings of the International Conference on Artificial Neural Networks, 2012, pp. 66-73.
[102] D.N.G. Silva, LD.S. Pacifico, T.B. Ludermir. An evolutionary extreme learning machine based on group search optimization, in: Proceedings of the Evolutionary Computation, 2011, pp. 574-580.
[103] P. Mohapatra, S. Chakravarty, P.K. Dash, An improved cuckoo search based extreme learning machine for medical data classification, Swarm Evolut. Comput. 24 (2015) 25-49.
[104] Y.S. Zhang, J. Wu, Z.H. Cai, P. Zhang, L. Chen, Memetic extreme learning machine, Pattern Recognit 58 (C) (2016) 135-148.
[105] R. Dash, P.K. Dash, R. Bisoi, A self adaptive differential harmony search based optimized extreme learning machine for financial time series prediction, Swarm Evolut, Comput, 19 (2014) 25-42
[106] Y.W. Huang, D.H. Lai, Hidden node optimization for extreme learning machine, Aasri Prroc. 3 (3) (2012) 375-380.

[107] A.S. Alencar, A.R.R. Netoa, J.P.P. Gomes, A new pruning method for extreme learning machines via genetic algorithms, Appl. Soft Comput. (2016) 101-107.
[108] B.X. Xue, X. Ma, J. Gu, Y.B. Li, An improved extreme learning machine based on variable-length particle swarm optimization, in: Proceedings of the IEEE International Conference on Robotics and Biomimetics, 2013, pp. 1030-1035.
[109] B.X. Xue, X. Ma, H.B. Wang, J. Gu, Y.B. Li, Improved variable-length particle swarm optimization for structure-adjustable extreme learning machine, Control Intell. Syst. 42 (4) (2014) 1-9.
[110] S.E. Fahlman, C. Lebiere, The cascade-correlation learning architecture, Adv. Neural Inf. Process. Syst. 2 (6) (1990) 524-532.
[111] G.B. Huang, L. Chen, CK. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (2006) 879-892.
[112] G.R. Feng, G.B. Huang, Q.P. Lin, R. Gay, Error minimized extreme learning machine with growth of hidden nodes and incremental learning, IEEE Trans. Neural Netw. 20 (8) (2009) 1352-1357.
[113] Y.B. Ye, Y. Qin, QR factorization based incremental extreme learning machine with growth of hidden nodes, Pattern Recognit. Lett. 65 (2015) 177-183.
[114] Y.M. Yang, Y.N. Wang, X.F. Yuan, Parallel chaos search based incremental extreme learning machine, Neural Process. Lett, 37 (3) (2013) 277-301.
[115] F. Han, M.R. Zhao, J.M. Zhang, Q.H. Ling. An improved incremental constructive single-hidden-layer feedforward networks for extreme learning machine based on particle swarm optimization, Neurocomputing 228 (2017) 133-142.
[116] F. Han, M-R. Zhao, J.-M. Zhang. An improved incremental error minimized extreme learning machine for regression problem based on particle swarm optimization, in: Advanced Intelligent Computing Theories and Applications, 2015, pp. 94-100.
[117] Q.W. Li, F. Han, Q.H. Ling. An improved double hidden-layer variable length incremental extreme learning machine based on particle swarm optimization, in: Proceedings of the International Conference on Intelligent Computing, 2018, pp. 1-10.
[118] C. Qian, Y. Yu, Z.H. Zhou, Pareto ensemble pruning, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2935-2941.
[119] D.H. Wang, M. Alhamdoosh, Evolutionary extreme learning machine ensembles with size control, Neurocomputing 102 (102) (2013) 98-110.
[120] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123-140.
[121] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression; a statistical view of boasting. Ann. Stat. 28 (2) (2000) 337-374.
[122] L. Breiman, Stacked regressions, Mach. Learn. 24 (1) (1996) 49-64.
[123] Z.H. Zhou, J.X. Wu, Y. Jiang, S.F. Chen, Genetic algorithm based selective neural network ensemble, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2001, pp. 797-802.
[124] Z.H. Zhou, J.X. Wu, W. Tang, Ensembling neural networks: many could be better than all, Artif. Intell. 137 (1) (2002) 239-263.
[125] LK. Hansen, P. Salamon, Neural network ensembles, IEEE Trans. Pattern Anal. Mach. Intell. 12 (10) (2002) 993-1001.

[126] LI. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Mach. Learn. 51 (2) (2003) 181-207.
[127] Y.S. Kim, W.N. Street, F. Menczer, Meta-evolutionary ensembles, in: Proceedings of the International Joint Conference on Neural Networks, 2002, pp. 2791-2796.
[128] Y. Lan, Y.C. Soh, G.B. Huang, Ensemble of online sequential extreme learning machine, Neurocomputing 72 (13) (2009) 3391-3395.
[129] N. Liu, H. Wang, Ensemble based extreme learning machine, IEEE Signal Process. Lett. 17 (8) (2010) 754-757.
[130] T.P.F.D. Lima, T.B. Ludermir. Ensembles of evolutionary extreme learning machines through differential evolution and fitness sharing, in: Proceedings of the International Joint Conference on Neural Networks, 2014, pp. 2677-2682
[131] J.FL de Oliveira, T.B. Ludermir, Homogeneous ensemble selection through hierarchical clustering with a modified artificial fish swarm algorithm, in: Proceedings of the IEEE International Conference on Tools with Artificial Intelligence, 2011, pp. 177-180.
[132] Y. Zhang, B. Liu, F. Yang, in: Differential Evolution Based Selective Ensemble of Extreme Learning Machine, IEEE, 2016a, pp. 1327-1333. Trustcom/bigdatase/ispa.
[133] Y. Zhang. B. Liu, J. Cai, S.H. Zhang, Ensemble weighted extreme learning machine for imbalanced data classification based on differential evolution, Neural Comput. Appl. 28 (1) (2016b) 1-9.
[134] Y. Liu, B. He, D. Dong, Y. Shen, T. Yan, N. Rui, A. Lendasse, Particle swarm optimization based selective ensemble of online sequential extreme learning machine, Math. Probl. Eng. 2015 (2015) 1-10.
[135] Q.H. Ling, Y.Q. Song, F. Han, D. Yang, D.S. Huang, An improved ensemble of random vector functional link networks based on particle swarm optimization with double optimization strategy, PLOS ONE 11 (11) (2016) e0165803, doi:10.1371/journal.pone.0165803.
[136] G.C. Luh, C.Y. Lin, Optimal design of truss-structures using particle swarm optimization, Comput. Struct. 89 (23) (2011) 2221-2232.
[137] D. Yang, F. Han, An improved ensemble of extreme learning machine based on attractive and repulsive particle swarm optimization, in: Proceedings of the International Conference on Intelligent Computing, 2014, pp. 213-220.
[138] Y.Q, Wu, F. Han, Q.H. Ling, An improved ensemble extreme learning machine based on ARPSO and tournament-selection, in: Proceedings of the International Conference in Swarm Intelligence, 2016, pp. 89-96.
[139] F. Han, D. Yang, Q.H. Ling. D.S. Huang. A novel diversity-guided ensemble of neural network based on attractive and repulsive particle swarm optimization, in: Proceedings of the International Joint Conference on Neural Networks, 2015, pp. 1-7.
[140] X.H. Zhu, ZW. Ni, M.Y. Cheng, F.F. Jin, J.M. Li, G. Weckman, Selective ensemble based on extreme learning machine and improved discrete artificial fish swarm algorithm for haze forecast, Appl. Intell. (2017) 1-19.
[141] C.M. Fonseca, P.J. Fleming, An overview of evolutionary algorithms in multiobjective optimization, Evolut. Comput. 3 (1) (1995) 1-16.

[142] A. Trivedi, D. Srinivasan, K. Sanyal, A. Ghosh, A survey of multiobjective evolutionary algorithms based on decomposition, IEEE Trans. Evolut. Comput. 21 (3) (2017) 440-462.
[143] D.A. Van Veldhuizen, G.B. Lamont, Multiobjective evolutionary algorithms: analyzing the state-of-the-art, Evolut. Comput., 8 (2) (2000) 125-147.
[144] LM. Antonio, CA.C. Coello, Coevolutionary multi-objective evolutionary algorithms: a survey of the state-of-the-art, IEEE Trans. Evolut. Comput. PP (99) (2017) 1-16.
[145] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2012, pp. 315-323.
[146] 1. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016. http://www.deeplearningbook.org
[147] J.D. Knowles, D.W. Corne, Approximating the nondominated front using the Pareto archived evolution strategy, Evolut. Comput. 8 (2) (2000) 149-172.
[148] CR. Raquel, P.C. Naval, An effective use of crowding distance in multiobjective particle swarm optimization, in: Proceedings of the Seventh Annual Conference on Genetic and Evolutionary Computation, 2005, pp. 257-264.
[149] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evolut. Comput. 6 (2) (2002) 182-197.
[150] Y.M. Cai, X.B. Liu, Y. Wu, P. Hu, R.L. Wang. B. Wu, Z.H. Cai, Extreme learning machine based on evolutionary multi-objective optimization, in: Proceedings of the International Conference on Bio-Inspired Computing: Theories and Applications, 2017, pp. 420-435.
[151] J. Echanobe, LD. Campo, V. Martnez, K. Basterretxea. Genetic algorithm-based optimization of ELM for on-line hyperspectral image classification, in: Proceedings of the International Joint Conference on Neural Networks, 2017, pp. 4202-4207.
[152] W. Du, S.Y.S. Leung, C.K. Kwong. Time series forecasting by neural networks: a knee point-based multiobjective evolutionary algorithm approach, Expert Syst. Appl. 41 (18) (2014) 8049-8061.
[153] CA.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Trans. Evolut. Comput. 8 (3) (2004) 256-279.
[154] W.T. Mao, M. Tian, X.Z. Cao, J.C. Xu, Model selection of extreme learning machine based on multi-objective optimization, Neural Comput. Appl. 22 (3-4) (2013) 521-529.
[155] VL Huang, P.N. Suganthan, J.J. Liang. Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems, Int. J. Intell. Syst. 21 (2) (2006) 209-226.
[156] G.C. Cawley, N.L.C. Talbot, Preventing over-fitting during model selection via Bayesian regularisation of the hyper-parameters, J. Mach. Learn. Res. (2007) 841-861.
[157] D. Lahoz, B. Lacruz, P.M. Mateo, A multi-objective micro genetic ELM algorithm. Neurocomputing 111 (6) (2013) 90-103.
[158] C. Liu, D.L. Chen, F.C. Wan, Multiobjective learning algorithm based on membrane systems for optimizing the parameters of extreme learning machine. Optik 127 (4) (2016) 1909-1917.

[159] B. Liu, Q.F. Zhang, G.G.E. Gielen, A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems, IEEE Trans. Evolut. Comput. 18 (2) (2014) 180-192.
[160] CL. Sun, Y.C. Jin, R. Cheng, J.L. Ding. J.C. Zeng. Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems, IEEE Trans. Evolut. Comput. 21 (4) (2017) 644-660.
[161] T. Chugh, K. Sindhya, J. Hakanen, K. Miettinen, A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms, Soft Comput. (2017) 1-30.
[162] J. Chen, J. Li, B, Xin, DMOEA-EC: decomposition-based multi-objective evolutionary algorithm with the e-constraint framework, IEEE Trans. Evolut. Comput. 21 (5) (2017) 714-730.
[163] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part 1: solving problems with box constraints, IEEE Trans. Evolut. Comput. 18 (4) (2014) 577-601.
[164] M.G. Gong, J. Liu, H. Li, Q. Cai, L.Z. Su, A multiobjective sparse feature learning model for deep neural networks, IEEE Trans. Neural Netw. Learn. Syst. 26 (12) (2017) 3263-3277.
[165] E. Real, S. Moore, A. Selle, S. Saxena, Y.L Suematsu, J. Tan, Q. Le, A. Kurakin, Large-scale evolution of image classifiers, in: Proceedings of the International Conference on Machine Learning, 2017, pp. 1-18.
[166] J.X. Tang, C.W. Deng, G.B. Huang. Extreme learning machine for multilayer perceptron, IEEE Trans. Neural Netw. Learn. Syst. 27 (4) (2016) 809-821.
[167] K. Sun, J.S. Zhang, C.X. Zhang, J.Y. Hu, Generalized extreme learning machine autoencoder and a new deep neural network, Neurocomputing 230 (2017) 374-381.
[168] T. Salimans, J. Ho, X. Chen, S. Sidor, I. Sutskever, Evolution strategies as a scalable alternative to reinforcement learning, 2017. arXiv: 1703.03864v2.
[169] N. Hansen. The CMA Evolution Strategy: a tutorial, 2016. arXiv: 1604.00772v1.
[170] C.H. Zheng, D.S. Huang, L. Shang, Feature selection in independent component subspace for microarray data classification, Neurocomputing 69 (1618) (2006) 2407-2410.
[171] Y. Zhang, B. Liu, J. Cai, S. Zhang, Ensemble weighted extreme learning machine for imbalanced data classification based on differential evolution, Neural Comput. Appl. 28 (1) (2016) 1-9.
[172] K.H. Liu, D.S. Huang, Cancer classification using rotation forest, Computers in Biology and Medicine 38 (5) (2008) 601-610.

Bölüm 2

SAHTE HABER TESPİTİ İÇİN MAKİNE ÖĞRENMESİ YÖNTEMLERİ

Ertuğrul AVCI[1]
Erol TERZİ[2]

1 Evidea – İş Zekası ve Raporlama Yöneticisi,
ertugrul_avci@icloud.com
ORCID ID:
2 Prof.Dr. Ondokuz Mayıs Üniversitesi,
eroltrz@omu.edu.tr
ORCID ID: 0000-0002-2309-827X

1. GİRİŞ

1.1. Problem Tanımı

İnternetin ve dijital medyanın yaygınlaşmasıyla birlikte bilgiye erişim kolaylaşmış, ancak bu gelişmeler beraberinde ciddi sorunları da getirmiştir. Bu sorunların başında sahte haberlerin yaygınlaşması gelmektedir. Sahte haberler, bireylerin yanlış yönlendirilmesine, kamuoyunun manipüle edilmesine ve toplumsal kutuplaşmanın artmasına neden olabilmektedir. Özellikle sosyal medya platformlarında yayılan sahte içerikler, doğruluğu teyit edilmeden geniş kitlelere ulaşabilmekte ve ciddi toplumsal sonuçlar doğurabilmektedir (Allcott & Gentzkow, 2017).

1.2. Çalışmanın Amacı

Bu çalışmanın temel amacı, metin madenciliği ve makine öğrenmesi yöntemleri kullanarak sahte haberleri tespit edebilen bir model geliştirmektir. Bu bağlamda, doğal dil işleme (NLP) teknikleri kullanılarak haber metinlerinden anlamlı özellikler çıkarılacak ve çeşitli sınıflandırma algoritmaları yardımıyla bu haberlerin doğruluğu değerlendirilecektir.

Proje kapsamında kullanılan veri setleri, “Fake.csv” (sahte haberler) ve “True.csv” (gerçek haberler) adlı iki ayrı dosyadan oluşmaktadır. Bu veri setleri üzerinden yapılacak analizlerle sahte haber tespiti probleminin farklı boyutları ele alınacaktır.

1.3. Sahte Haber Nedir?

Sahte haber (Fake News), yanlış, yanıltıcı veya tamamen uydurma bilgileri içeren, genellikle belirli bir ajandaya hizmet eden haber türüdür. Sahte haberler çoğunlukla propaganda, siyasi manipülasyon, finansal çıkar sağlama gibi motivasyonlarla oluşturulmaktadır. Bu tür haberler, güvenilir haber kaynaklarının görünümünü taklit ederek okuyucunun inancını kazanmayı amaçlar (Lazer et al., 2018).

1.4. Sahte Haberlerin Toplum Üzerindeki Etkileri

Sahte haberlerin toplum üzerindeki etkileri oldukça ciddidir. Seçim süreçlerinin manipüle edilmesi, bireylerin yanlış sağlık bilgileriyle yönlendirilmesi ve sosyal çatışmaların körüklenmesi bu etkilerden sadece birkaçıdır. Özellikle COVID-19 pandemisi süresince yanlış bilgilendirme ve dezenformasyonun etkileri net bir şekilde görülmüştür (Zarocostas, 2020).

1.5. Makine Öğrenmesi ile Sahte Haber Tespiti

Makine öğrenmesi, büyük veri kümeleri üzerinde öğrenme sağlayarak tahmin ve sınıflandırma görevlerinde oldukça etkili sonuçlar sunmaktadır. Bu bağlamda, sahte haber tespiti gibi metin sınıflandırma problemlerinde de çeşitli algoritmalar (Naive Bayes, SVM, Random Forest, LSTM vb.) yaygın

olarak kullanılmaktadır. Bu çalışmada da farklı makine öğrenmesi modelleri test edilerek en yüksek doğruluk oranı elde edilmeye çalışılacaktır.

1.6. Çalışmanın Katkısı ve Önemi

Bu proje, sahte haber tespiti alanında hem akademik hem de pratik açıdan değerli bir katkı sunmayı amaçlamaktadır. Elde edilecek bulgular, medya kuruluşları, sosyal medya platformları ve araştırmacılar için önemli içgörüler sağlayabilir. Ayrıca, bu çalışma sayesinde Türkçe haberlerde kullanılabilecek güvenilir bir model altyapısı oluşturulması da hedeflenmektedir.

2. LİTERATÜR TARAMASI

2.1. Sahte Haber Tespiti Üzerine Genel Yaklaşımlar

Sahte haber tespiti alanı, son yıllarda doğal dil işleme (NLP) ve makine öğrenmesi tekniklerinin gelişmesiyle birlikte oldukça hızlı ilerlemiştir. Yapılan çalışmalar çoğunlukla üç ana yaklaşım etrafında toplanmaktadır:

· İçerik Tabanlı Yöntemler: Haber metninin dil yapısı, sözcük frekansları, başlık analizi ve metin karmaşıklığı gibi içerik özelliklerine dayanır. TF-IDF, bag-of-words (BoW), LDA gibi teknikler sıkça kullanılmaktadır.

· Kaynak ve Meta Veri Tabanlı Yöntemler: Haberi paylaşan yazar, kurum, tarih, paylaşım sıklığı gibi meta veriler analiz edilir.

· Sosyal Etkileşim Tabanlı Yöntemler: Özellikle sosyal medya verileri üzerinde yapılan çalışmalarda haberin ne şekilde yayıldığı, kimler tarafından ve ne hızda paylaşıldığı gibi veriler incelenir.

2.2. Doğal Dil İşleme (NLP) Yöntemleri ile Sahte Haber Tespiti

NLP teknikleri, sahte haber tespitinde anahtar rol oynamaktadır. En çok kullanılan teknikler:

· N-gram modelleri

· TF-IDF (Term Frequency-Inverse Document Frequency)

· Word2Vec, GloVe gibi kelime gömme (embedding) teknikleri

· Sentiment Analysis (Duygu Analizi)

· Topic Modeling (LDA, NMF)

Bu yöntemler haber metinlerinin anlamını sayısal temsillere dönüştürerek sınıflandırma algoritmalarına girdi sağlar.

2.3. Makine Öğrenmesi Modelleri ile Sahte Haber Sınıflandırması

Makine öğrenmesi algoritmaları, sahte haberleri sınıflandırmak için güçlü araçlardır. Aşağıdaki algoritmalar yaygın olarak kullanılmaktadır:

· Naive Bayes (NB): Hızlı ve basit yapısı nedeniyle metin sınıflandır-

mada ilk sıralarda tercih edilir.

· Destek Vektör Makineleri (SVM): Özellikle yüksek boyutlu verilerde iyi performans gösterir.

· Karar Ağaçları ve Random Forest: Açıklanabilirlik açısından avantajlıdır.

· Lojistik Regresyon: İkili sınıflandırma problemleri için temel yöntemdir.

· XGBoost / LightGBM: Gelişmiş ağaç tabanlı boosting algoritmalarıdır.

2.4. Derin Öğrenme ve Hibrit Yaklaşımlar

Son yıllarda sahte haber tespiti için **derin öğrenme modelleri** yaygınlaşmıştır. Bu modellerin bazıları:

· Convolutional Neural Networks (CNN): Metin içindeki yapısal kalıpları öğrenmede etkilidir.

· Recurrent Neural Networks (RNN) / LSTM: Zaman sıralı metin verilerinde daha başarılıdır.

· Transformer tabanlı modeller (BERT, RoBERTa): Konu bağlamını anlamada üstündür.

Hibrit modeller ise içerik + sosyal ağ verilerini birleştirerek daha yüksek başarı sağlar.

2.5. Sahte Haber Tespiti Üzerine Türkçe Çalışmalar

Türkçe veri setleri ve çalışmalar daha sınırlı olmakla birlikte giderek artmaktadır. Bazı örnek çalışmalar şunlardır:

· Habernet, trFakeNews, Türkçe haber başlıklarıyla oluşturulmuş özel veri setleri

· Türkçe dil yapısına uygun morfolojik analiz, Zemberek NLP kütüphanesi vb. yöntemler kullanılmıştır.

2.6. Zorluklar ve Açık Problemler

Sahte haber tespiti her ne kadar günümüz teknolojileriyle çözülebilir gibi görünse de hâlâ birçok teknik, etik ve yapısal zorluk barındırmaktadır. Bu bölümde karşılaşılan temel problemler, çözüm yönelimleri ve araştırmaya açık alanlar detaylandırılmaktadır.

2.6.1. Veri Seti Dengesizliği (Imbalanced Dataset Problem)

Çoğu sahte haber veri setinde gerçek ve sahte haberler eşit sayıda bulunmaz, bu da modellerin öğrenme sürecinde yanlı tahminler yapmasına yol

açar. Gerçek haberler genellikle daha fazladır, bu da modelin "gerçek" sınıfına doğru eğilmesine sebep olur. Bu dengesizliğe çözüm olması adına aşağıda yer alan teknikleri kullanabiliriz.

· SMOTE (Synthetic Minority Oversampling Technique) gibi yöntemlerle azınlık sınıfın örnek sayısı artırılabilir.

· Ağırlıklı kayıp fonksiyonları, örneğin class_weight='balanced' gibi hiperparametrelerle model eğitimi optimize edilebilir.

2.6.2. Veri Doğruluğu ve Etiketleme Problemleri

Haberlerin doğru şekilde "sahte" ya da "gerçek" olarak etiketlenmesi oldukça zordur. Hatalı etiketlemeler modelin hatalı öğrenmesine neden olabilir. Ayrıca, etiketteki sübjektiflik (örneğin, abartılı başlık mı, yoksa sahte içerik mi?) de değerlendirme süreçlerini karmaşıklaştırır.

2.6.3. Dilsel Karmaşıklık ve Manipülasyon

Sahte haberler genellikle oldukça ikna edici bir dille yazılır. Duygu sömürüsü, abartma, retorik sorular ve görsel desteklerle okur etkilenir. Bu durum, metin tabanlı NLP modelleri için ciddi bir yanıltıcı faktördür.

· Zorluk: Sadece kelime frekanslarına ya da cümle yapısına dayalı modeller bu karmaşıklığı çözemeyebilir.

· İleri Seviye NLP Gereksinimi: Transformer tabanlı modeller (BERT, GPT vb.) bu dili daha iyi analiz edebilir.

2.6.4. Model Açıklanabilirliği (Explainability)

Sahte haber tespiti gibi hassas kararların verildiği sistemlerde "neden bu karar verildi?" sorusunun yanıtı önemlidir. Ancak, özellikle derin öğrenme modelleri "kara kutu" (black box) olarak çalıştığı için kararları açıklamak güçtür.

· XAI (Explainable AI) teknikleri, LIME, SHAP gibi yöntemlerle model kararlarının görselleştirilmesi sağlanabilir.

· Etik Sorumluluk: Haberin sahte olduğu iddiası, bir kişinin veya kurumun itibarını etkileyebilir. Bu nedenle açıklanabilirlik, etik değerlendirmelerle birlikte düşünülmelidir.

2.6.5. Çok Dilli (Multilingual) ve Kültürel Farklılıklar

Sahte haber tespiti çoğunlukla İngilizce metinler üzerinden geliştirilmiştir. Ancak farklı dillerde (örneğin Türkçe, Arapça, Çince) dil yapısı, deyimler, gramer kuralları ve söylem biçimi değişiklik gösterir.

· Türkçede eklemeli yapı, sözcük köklerinin değişmesi, mecaz kullanımı gibi öğeler dil modeli geliştirmeyi zorlaştırır.

· Dil bağımsız evrensel modeller hâlen araştırma aşamasındadır.

2.6.6. Görsel ve Çok Modlu Haberler

Bazı sahte haberler yalnızca metin değil; görseller, videolar ve bağlantılarla birlikte servis edilir. Metin odaklı modeller bu bağlamı göz ardı edebilir. Bu nedenle multimodal modellerin geliştirilmesi gerekir.

· EANN (Event Adversarial Neural Network) gibi çok modlu derin öğrenme yapıları bu alanda umut vadetmektedir.

· Görseldeki manipülasyonları algılayan modeller sahte içeriği daha doğru yakalayabilir.

2.6.7. Gerçek Haberler de Yanıltıcı Olabilir

Gerçek haberler bile önyargılı, eksik ya da yanlış yönlendirmeli olabilir. Sahte haber olarak sınıflandırılmasalar da okuyucunun algısını yanlış şekillendirebilirler. Bu gri alanlar sınıflandırmayı daha karmaşık hâle getirir.

2.6.8. Gerçek Zamanlı Sahte Haber Tespiti

Gerçek zamanlı sahte haber tespiti hâlâ büyük bir açık alandır. Sosyal medya gibi hızlı platformlarda haberlerin yayılma hızı çok yüksek olduğundan, modellerin güncel ve hızlı olması gerekir.

· Streaming veri modelleri

· Zaman serileriyle birlikte çalışan NLP algoritmaları

3. VERİ SETİ TANITIMI ve KEŞİFSEL VERİ ANALİZİ (EDA)

3.1. Veri Setlerinin Tanıtımı

Bu projede elimizde iki tane önemli veri seti var: Fake.csv ve True.csv. İsimlerinden de anlaşılacağı gibi, biri sahte haberleri, diğeri ise gerçek haberleri içeriyor.

Tablo 3.1. *Veri Seti Tanıtımı*

Veri Seti	Kayıt Sayısı	Sütun Sayısı
Fake.csv (Sahte Haber)	23481	5
True.csv (Sahte Haber)	21417	5
Toplam	44898	5

Sahte Haberler hakkında bilgiler aşağıda yer almaktadır.

Bu dosyada toplamda 23.481 tane haber kaydı var. Her bir satır bir haber anlamına geliyor. Her haber için şu bilgiler yer alıyor:

· title: Haberin başlığı

· text: Haberin tam metni

· subject: Haberin hangi konuyla ilgili olduğu (örneğin politika, dünya haberleri vs.)

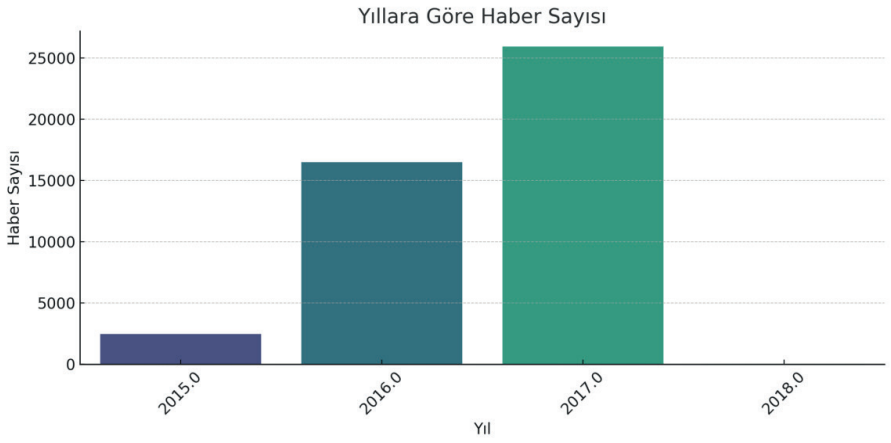
· date: Haberin yayınlandığı tarih

Bu haberler genellikle gerçekmiş gibi gösterilen ama aslında doğruluğu olmayan bilgiler içeriyor. Sahte haberlerin nasıl bir yapıya sahip olduğunu anlamak için bu veri çok işimize yarayacak.

Gerçek Haberler hakkında bilgiler aşağıda yer almaktadır.

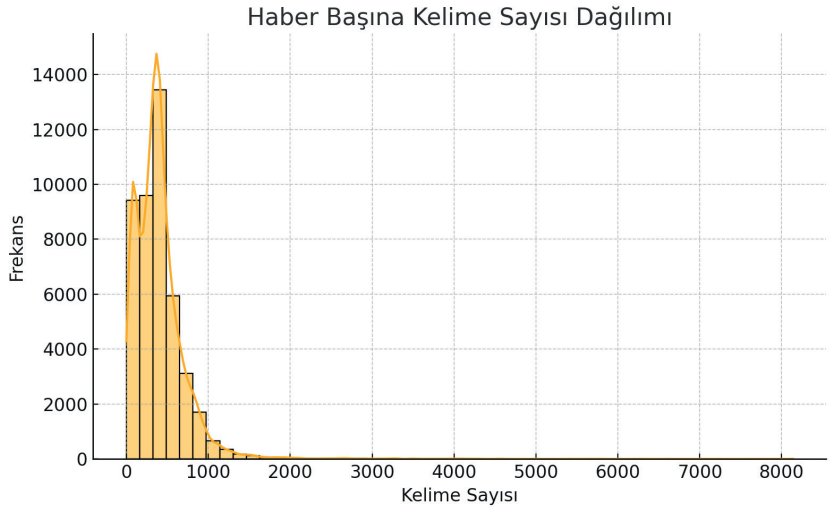
Bu dosyada da 21.417 tane gerçek haber bulunuyor. Yapısı Fake.csv ile aynı. Yani bu dosyada da başlık, metin, konu ve tarih bilgileri yer alıyor.

Gerçek haberlerle sahte haberleri karşılaştırmak, aralarındaki dil farklılıklarını, başlık yapısını ve konularını incelemek için bu iki veri seti çok uygun.



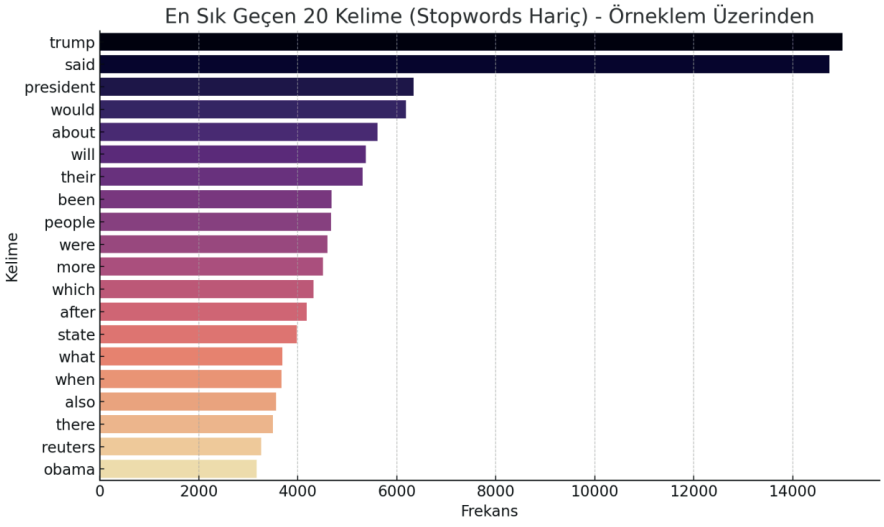
Şekil 3.1. Yıllara Göre Haber Sayısı

Bu grafikte yıllara göre yayımlanan haber sayıları gösterilmektedir. 2015-2017 yılları arasında belirgin bir artış olduğu görülmektedir. Bu durum, dijital haberleşmenin yaygınlaşmasıyla sahte ve gerçek haber üretiminin yoğunlaştığını göstermektedir.



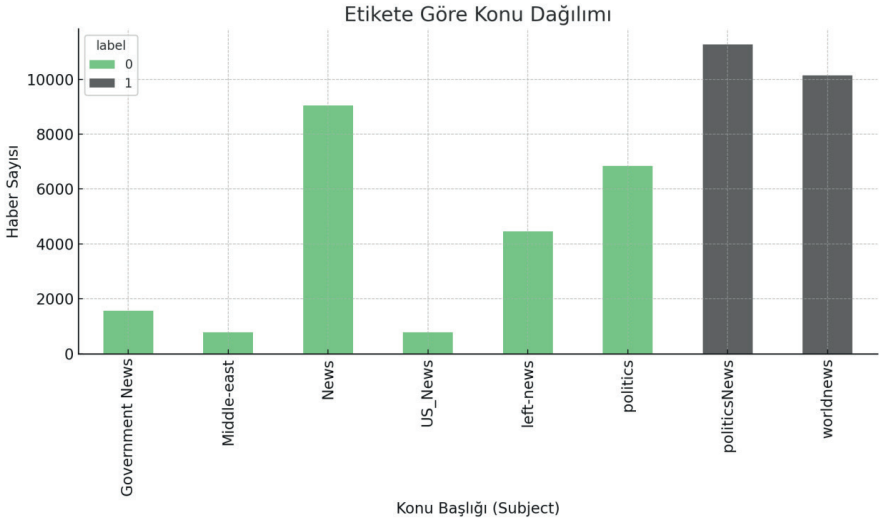
Şekil 3.2. Haber Başına Kelime Sayısı Dağılımı

Haber metinlerinin içerik uzunluklarını gösteren bu histogramda, çoğu haberin 200-800 kelime arasında olduğu gözlemlenmiştir. Dağılım simetrik bir yapıya sahiptir ve bazı haberlerin oldukça uzun içerikler barındırdığı dikkat çekmektedir.



Şekil 3.3. En Sık Geçen 20 Kelime

5000 örnek üzerinden yapılan analizde, en sık geçen kelimeler belirlenmiştir. “trump”, “said”, “people”, “obama” gibi kelimeler ön plana çıkmaktadır. Bu durum, haberlerin çoğunlukla politik içerikli olduğunu ve belirli figürler etrafında şekillendiğini göstermektedir.



Şekil 3.4. Etikete Göre Konu Dağılımı

Şekil 3.5. Zaman Serisi Analizi

Günlük bazda haber üretim miktarını gösteren bu zaman serisi grafiğinde, bazı dönemlerde ani artışlar gözlemlenmektedir. Bu artışlar, önemli siyasi gelişmeler veya kriz dönemlerine denk geliyor olabilir. Grafik, sahte ve gerçek haberlerin zaman içinde nasıl yayıldığını gözlemlemek adına önemli ipuçları sunmaktadır.

3.2. Veri Birleştirme ve Etiketleme

Projede işlerimizi kolaylaştırmak adına bu iki dosyayı birleştirdik. Böylece elimizde toplamda 44.898 haber bulunan tek bir veri seti oluştu. Bu birleşik veri setine her haberin "gerçek" mi yoksa "sahte" mi olduğunu belirten bir etiket ekledik:

- Sahte haberlere "0"
- Gerçek haberlere "1" etiketi verdik.

Bu sayede makine öğrenmesi algoritmalarına bu veriyi kolayca öğretebileceğiz.

3.3. Eksik Değer Analizi

Projede kullanmış olduğumuz veri setlerinde eksik değer bulunmamaktadır. Tüm satırlar doludur. Ancak tarih (date) sütunu bazı analizlerde NaT (geçersiz tarih) formatına dönüşebilir; bu nedenle dikkatle işlenmelidir.

4. VERİ ÖN İŞLEME ve ÖZELLİK MÜHENDİSLİĞİ

Bu aşamada, elimizdeki haber metinlerini daha anlamlı hale getirebilmek için biraz temizlik ve sayısal hale dönüştürme işlemleri yapmamız gerekiyor. Yani bilgisayarın anlayacağı forma sokuyoruz.

4.1. Metin Temizleme

İlk işimiz, haberlerin içeriğindeki gereksiz karakterlerden ve gürültülerden kurtulmak.

Tablo 4.1. *Veri Ön İşleme Adımları*

Adım	İşlem	Açıklama
1	Küçük harfe çevirme	Tüm metinler lowercase hale getirildi.
2	Noktalama İşaretlerini temizleme	String.punctuation karakterleri kaldırıldı.
3	Sayıları ve sayısal ifadeleri kaldırma	Sayısal içerikler metinden temizlendi.
4	TF-IDF dönüşümü (5000 özellik)	TF_IDF ile sayısal öznitelikler çıkartıldı.
5	Eğitim/Test ayrımı (%80/%20)	Train_test_split kullanılarak bölündü

Örnek verecek olursak eğer:

· Büyük harfleri küçük harfe çevirdik.

· Noktalama işaretlerini sildik (.,!? gibi).

· Web adresleri (örneğin “https://...”) temizlendi.

· Satır sonları ve sayı içeren kelimeler çıkartıldı.

Bu işlemlerden sonra daha sade ve işlenebilir metinler elde ettik.

4.2. Sayısallaştırma (TF-IDF Vektörleştirme)

Makine öğrenmesi modelleri yazıları anlayamaz. O yüzden metinleri sayısal vektörlere dönüştürmemiz lazım. Bu iş için “TF-IDF” adlı yöntemi kullandık.

Peki TF-IDF yöntemi ile neler yapılabilir:

· Her kelimenin metindeki sıklığını ölçer.

· Ama bu kelime başka tüm metinlerde de çok geçiyorsa (örneğin “the”, “and”), puanını düşürür.

Böylece model sadece anlamlı kelimelere odaklanır.

Biz de her haber metninden en fazla 5000 tane anlamlı kelime çıkartıp bunları sayısal vektörlere dönüştürdük.

4.3. Eğitim ve Test Ayrımı

Veriyi modele öğretmeden önce:

- %80'ini eğitim verisi olarak ayırdık.
- %20'sini test için sakladık.

Yani model önce öğreniyor, sonra hiç görmediği haberlerle sınava giriyor. Bu sayede ne kadar doğru tahmin yaptığını test edebileceğiz.

5. MODELLEME ve ALGORİTMALARIN EĞİTİMİ

Çalışmanın bu bölümünde, sahte haber tespiti için kullanılan veri setinin hazırlanması ve makine öğrenmesi modellerinin eğitimi süreci ayrıntılı olarak açıklanmıştır.

Kullanılan verinin eğitim ve test olarak bölünmesi, farklı algoritmalarla modelleme işlemlerinin gerçekleştirilmesi ve elde edilen sonuçların değerlendirilmesi amacıyla sistematik bir yaklaşım izlenmiştir.

5.1. Eğitim ve Test Veri Setinin Hazırlanması

Çalışmada kullanılan veri seti, sahte ve gerçek haberlerden oluşmaktadır. Öncelikle metin verileri temizlenmiş ve TF-IDF yöntemi kullanılarak haber metinlerinden sayısal öznitelikler çıkarılmıştır.

Tablo 5.1. *Kullanılan Makine Öğrenmesi Modelleri*

Model	Model Tipi	Avantajları	Dezavantajları
Logistic Regression	Doğrusal	Basit ve etkili, hızlı çalışır.	Doğrusal olmayan ayrımlarda yetersiz olabilir.
Naive Bayes	Olsaılık Tabanlı	Hafif ve düşük hesaplama maliyeti.	Dil karmaşıklığını iyi yönetemez.
Random Forest	Topluluk (Ensemble)	Yüksek doğruluk, overfitting'e karşı dirençli.	Eğitim süresi uzundur.
Support Vector Machine (SVM)	Doğrusal Ayırıcı	Yüksek boyutlu verilerde iyi performans.	Eğitim süresi yüksektir, kernel ayarları hassas.

Elde edilen veri seti, %80 eğitim ve %20 test olacak şekilde ikiye bölünmüştür. Bu bölünme ile modellerin, verinin büyük bir kısmı üzerinde öğrenmesi ve kalan kısım üzerinde sınanması sağlanmıştır.

5.2. Kullanılan Makine Öğrenmesi Modelleri

Çalışmada farklı yapısal özelliklere sahip dört makine öğrenmesi algoritması tercih edilmiştir:

- Logistic Regression (Lojistik Regresyon):

Doğrusal ayırımı temel alan ve ikili sınıflandırma problemleri için yaygın olarak kullanılan bir modeldir.

· Multinomial Naive Bayes:

Özellikle metin madenciliği uygulamalarında etkili, olasılık temelli bir sınıflayıcıdır.

· Random Forest Classifier:

Çok sayıda karar ağacının birleşimiyle oluşan, topluluk öğrenmesine dayalı bir algoritmadır.

· Support Vector Machine (SVM):

Yüksek boyutlu verilerde etkili çalışan ve sınıfları en iyi şekilde ayıran hiper düzlemleri bulan bir modeldir.

Bu modellerin seçilme nedeni, sahte haber tespiti gibi metin temelli sınıflandırma problemlerinde farklı öğrenme stratejilerinin başarılarını karşılaştırabilmektir.

5.3. Model Eğitimi

Hazırlanan eğitim verileri (X_train**,** y_train**)** kullanılarak her bir model ayrı ayrı eğitilmiştir. Eğitim sürecinde kullanılan temel parametreler şunlardır:

· Logistic Regression: max_iter=1000 ile optimize edilmiştir.

· Multinomial Naive Bayes: Standart hiperparametrelerle kullanılmıştır.

· Random Forest Classifier: n_estimators=100 (100 ağaç) seçilmiştir.

· SVM: Lineer çekirdek (LinearSVC) kullanılarak eğitilmiştir.

Her model, eğitim verilerinde sahte ve gerçek haberlerin karakteristik özelliklerini öğrenerek sınıflandırma yapacak şekilde eğitilmiştir.

6. MODEL DEĞERLENDİRME ve SONUÇLARIN YORUMLANMASI

Modelleme sürecinde farklı makine öğrenmesi algoritmaları kullanılarak sahte haber tespiti üzerine çeşitli deneyler gerçekleştirilmiştir.

Elde edilen modellerin başarısını objektif olarak ölçebilmek ve karşılaştırabilmek adına bazı performans değerlendirme metrikleri kullanılmıştır.

Bu bölümde, uygulanan modellerin performans kriterleri açıklanmakta ve elde edilen sonuçlar detaylı bir şekilde yorumlanmaktadır.

6.1. Değerlendirme Yöntemleri

Modellerin performansını değerlendirmek için aşağıdaki temel metrikler

kullanılmıştır:

Accuracy (Doğruluk Oranı): Modelin genel doğruluk seviyesi

Precision (Kesinlik): Pozitif tahminlerin doğruluğu

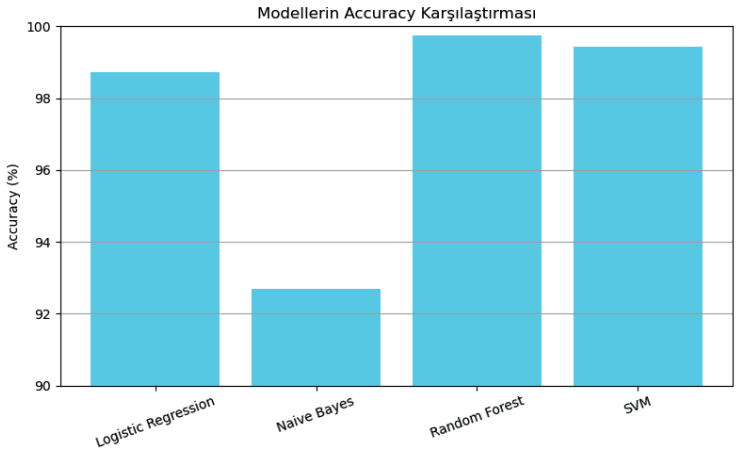
Recall (Duyarlılık): Gerçek pozitiflerin tespit oranı

F1-Score: Precision ve Recall'un harmonik ortalaması

Bu metrikler, sklearn kütüphanesinin classification_report fonksiyonu ile hesaplanmıştır.

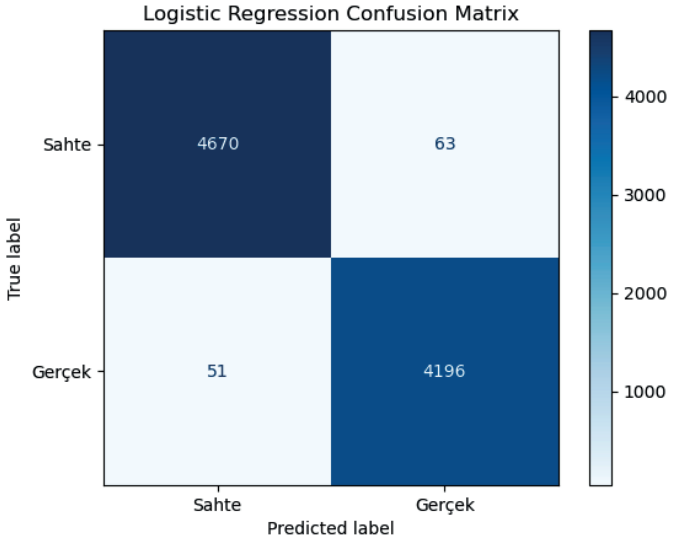
6.2. Test Sonuçları

Her model eğitim tamamlandıktan sonra, test veri seti (X_test) üzerinde sınanmıştır.



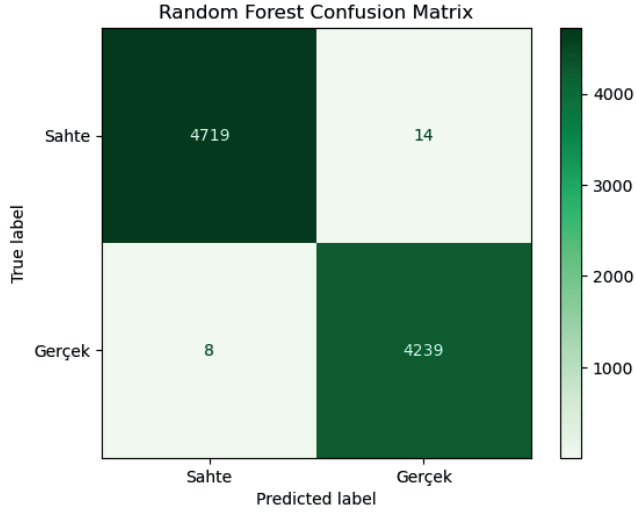
Şekil 6.1. *Modellerin Accuracy Karşılaştırması (Bar Chart)*

Dört farklı makine öğrenmesi algoritmasının test verisi üzerindeki doğruluk (accuracy) oranları karşılaştırmalı olarak sunulmuştur. Random Forest ve SVM modelleri en yüksek doğruluk oranlarına ulaşırken, Naive Bayes modeli diğer modellere göre daha düşük bir doğruluk sergilemiştir.



Şekil 6.2. *Logistic Regression Confusion Matrix*

Logistic Regression modeli için oluşturulan karışıklık matrisi (confusion matrix) gösterilmektedir. Model yüksek doğrulukla sınıflandırmış olmakla birlikte minimal yanlış sınıflamalar gözlemlenmiştir.



Şekil 6.3. *Random Forest Confusion Matrix*

Random Forest modeli için oluşturulan karışıklık matrisini göstermektedir. Random Forest modeli, sınıflandırmada neredeyse mükemmel bir başarı sergileyerek hata oranını minimuma indirmiştir. Elde edilen performans sonuçları aşağıda özetlenmiştir:

Tablo 6.1. *Model Karşılaştırma Tablosu*

Model	Accuracy (%)	Precision	Recall	F1-Score
Logistic Regression	98.73	0.99	0.99	0.99
Naive Bayes	92.70	0.93	0.93	0.93
Random Forest	99.75	1.00	1.00	1.00
SVM	99.42	0.99	0.99	0.99

6.3. Sonuçların Yorumu

Sonuçlar incelendiğinde, **Random Forest** modelinin %99,75 doğruluk oranı ile en yüksek başarıyı sağladığı görülmüştür.

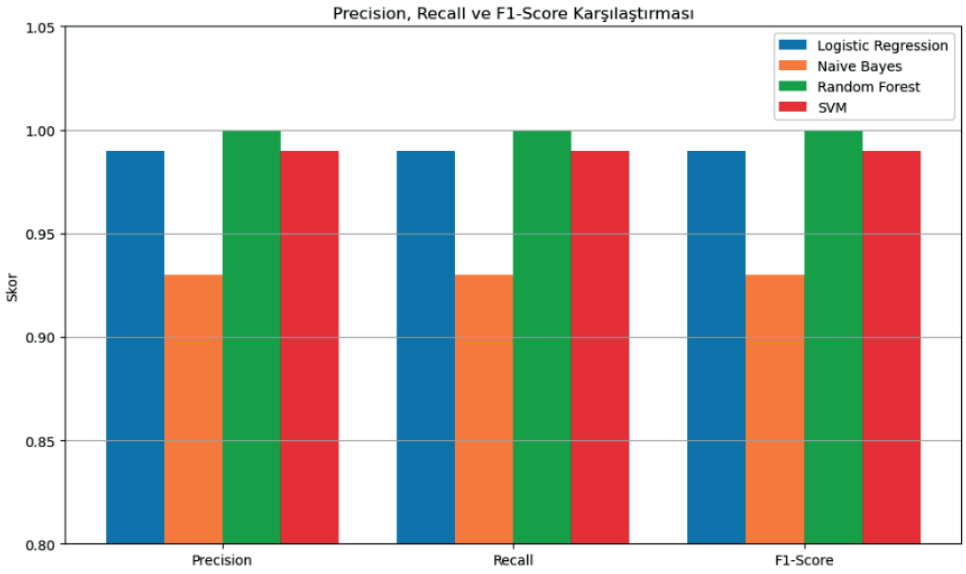
Random Forest modeli, aynı zamanda Precision, Recall ve F1-Score değerlerinde 1.00 gibi mükemmel skorlar elde etmiştir. Bu model hem doğruluk hem de diğer performans metriklerinde üstün başarı göstererek, sahte haber tespitinde topluluk öğrenmesi yöntemlerinin etkinliğini bir kez daha kanıtlamıştır.

SVM ve Logistic Regression modelleri de %99'un üzerinde doğruluk oranları ile sahte haber tespitinde etkili olmuştur.

Özellikle SVM modeli, dilin yapısal özelliklerini yüksek doğrulukla ayırarak etkileyici bir başarı göstermiştir.

Naive Bayes modeli ise %92,70 doğruluk oranı ile diğer modellere kıyasla daha düşük bir performans sergilemiştir.

Bu durum, Naive Bayes'in varsayımsal sadeleştirmeleri nedeniyle karmaşık dil yapılarında daha düşük başarı göstermesinden kaynaklanmaktadır.



Şekil 6.4. *Precision, Recall ve F1-Score Karşılaştırması*

Dört modelin Precision, Recall ve F1-Score değerleri karşılaştırılmaktadır. Random Forest modeli tüm metriklerde 1.00 değerine ulaşarak en üstün performansı sergilemiştir. Logistic Regression ve SVM modelleri de dengeli ve yüksek performans göstermiştir, Naive Bayes modeli ise diğer modellere göre daha düşük değerler üretmiştir.

Genel olarak, daha kompleks ve topluluk öğrenme tabanlı modellerin (örneğin Random Forest) sahte haber tespiti problemlerinde daha üstün sonuçlar verdiği sonucuna ulaşılmıştır.

7. SONUÇLAR ve TARTIŞMA

Bu çalışmada, sahte haber tespiti için çeşitli makine öğrenmesi algoritmaları uygulanmış ve elde edilen performans metrikleri ışığında modellerin etkinlikleri karşılaştırılmıştır.

Veri seti üzerinde gerçekleştirilen metin ön işleme adımları sonrasında TF-IDF yöntemi kullanılarak sayısal öznitelikler çıkarılmış ve dört farklı sınıflandırma algoritması eğitilmiştir: Logistic Regression, Naive Bayes, Random Forest ve Support Vector Machine (SVM).

Elde edilen sonuçlara göre, Random Forest modeli %99,75 doğruluk oranı ile en yüksek başarıyı göstermiştir. SVM ve Logistic Regression modelleri de sırasıyla %99,42 ve %98,73 doğruluk oranları ile güçlü bir performans sergilemiştir. Buna karşılık, Naive Bayes modeli %92,70 doğruluk oranı ile diğer modellere göre daha düşük bir başarı göstermiştir.

Modellerin precision, recall ve F1-Score değerleri de incelendiğinde, özellikle Random Forest ve SVM modellerinin sahte ve gerçek haberleri dengeli bir şekilde sınıflandırabildiği gözlemlenmiştir. Logistic Regression modeli ise basit yapısına rağmen etkili sonuçlar üretmiştir. Naive Bayes modeli, hızlı ve düşük maliyetli bir çözüm sunmasına karşın daha kompleks dil yapılarını ayrıştırmada sınırlı kalmıştır.

Bu bulgular, sahte haber tespitinde karmaşık model yapılarına sahip algoritmaların (özellikle topluluk öğrenme yöntemleri) daha etkili olduğunu göstermektedir. Ayrıca, doğru öznitelik çıkarımı ve iyi bir ön işleme süreci, model başarısına doğrudan etki eden kritik faktörler arasında yer almaktadır.

Çalışmanın genel sonuçları, makine öğrenmesi tabanlı yaklaşımların sahte haber tespitinde güçlü bir araç olduğunu desteklemekte ve ileride yapılacak geliştirmeler için sağlam bir temel sunmaktadır.

Bu çalışmada elde edilen başarılı sonuçlara rağmen, sahte haber tespiti gibi karmaşık ve dinamik bir problem için geliştirilebilecek pek çok alan bulunmaktadır. Yapılan analizler doğrultusunda gelecekte gerçekleştirilebilecek iyileştirme önerileri şu şekilde sıralanabilir:

Daha Büyük ve Çeşitlendirilmiş Veri Setleri Kullanımı : Çalışmada kullanılan veri seti belirli kaynaklardan derlenmiştir. Gelecekte, farklı dil, kültür ve kaynaklardan toplanmış daha büyük ölçekli veri setleri kullanılarak modellerin genelleme yetenekleri artırılabilir. Özellikle sosyal medya içerikleri, forum mesajları gibi kısa ve informal metinler de analize dahil edilebilir.

Derin Öğrenme Modellerinin Uygulanması : Geleneksel makine öğrenmesi algoritmalarının yanında, derin öğrenme tabanlı yöntemlerin (örneğin LSTM, BERT, RoBERTa gibi transformer tabanlı modeller) kullanılması, metin içerisindeki bağlam ve anlamsal ilişkilerin daha iyi öğrenilmesine katkı sağlayabilir. Bu sayede modelin dilin inceliklerini daha doğru yakalaması mümkün olur.

Çoklu Veri Modları (Multimodal) Analizi : Sadece metin bazlı analizlerin ötesine geçilerek, görsel ve video içeriklerin de dahil edildiği çoklu veri modlarına dayalı sahte haber tespit sistemleri geliştirilebilir. Özellikle haberlerde kullanılan görsellerin manipülasyonu, haberin güvenilirliğini etkileyen önemli faktörlerden biridir.

Gerçek Zamanlı Sahte Haber Tespiti : Geliştirilecek sistemlerin sosyal medya platformlarında veya haber portallarında gerçek zamanlı olarak sahte haber yayılımını tespit edebilmesi önemli bir araştırma alanıdır. Bu amaçla, hızlı ve düşük gecikmeli tahmin yapabilen algoritmalara ihtiyaç duyulmaktadır.

Açıklanabilir Yapay Zeka (Explainable AI) Yöntemlerinin Kullanımı : Özellikle sahte haber tespiti gibi kritik uygulamalarda modelin aldığı kararların anlaşılabilir olması önemlidir. Bu bağlamda, SHAP, LIME gibi açıklanabilir yapay zeka yöntemleri entegre edilerek model tahminlerinin nedenleri kullanıcıya sunulabilir. Bu, kullanıcı güvenliğini ve sistem şeffaflığını artırır.

KAYNAKLAR

Akçora, C. G., & Aydın, M. A. (2020). Türkçe sahte haber tespiti üzerine bir çalışma: Derin öğrenme yöntemleri ile karşılaştırmalı analiz. *Journal of Information Security Technologies*.

Ahmed, H., Traore, I., & Saad, S. (2017). Detecting opinion spams and fake news using text classification. *Security and Privacy*, 1(1), e9.

Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2), 211-236.

Baly, R., Karadzhov, G., Alexandrov, D., Glass, J., & Nakov, P. (2018). Predicting factuality of reporting and bias of news media sources. *arXiv preprint* arXiv:1810.01765.

Carvalho, P., Lima, H. S., Vieira, K., Meira Jr, W., & Almeida, V. (2011). Media bias in presidential elections: a semi-supervised learning approach. *Data Mining and Knowledge Discovery*, 23(3), 587–614.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.

Çeliktutan, O., & Yıldız, O. T. (2019). Sahte haber tespiti için Türkçe veri seti oluşturulması ve değerlendirilmesi. *IEEE 27th Signal Processing and Communications Applications Conference (SIU)*.

Conneau, A., & Lample, G. (2019). Cross-lingual language model pretraining. *NeurIPS*.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint* arXiv:1810.04805.

Lazer, D. M., Baum, M. A., Grinberg, N., Friedland, L., Joseph, K., & Hobbs, W. R. (2018). The science of fake news. *Science*, 359(6380), 1094-1096.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *NeurIPS*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint* arXiv:1301.3781.

Nakov, P., et al. (2018). Automated fact-checking for assisting human fact-checkers. *Proceedings of the IJCAI*.

Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., & Choi, Y. (2017). Truth of varying shades: Analyzing language in fake news and political fact-checking. *Proceedings of EMNLP*.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. *Proceedings of ACM SIGKDD*.

Ruchansky, N., Seo, S., & Liu, Y. (2017). CSI: A hybrid deep model for fake news detection. *Proceedings of CIKM 2017*.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1), 22-36.

Shu, K., Wang, S., & Liu, H. (2017). Understanding fake news: A survey. *arXiv preprint* arXiv:1708.01967.

Tandoc Jr, E. C., Lim, Z. W., & Ling, R. (2018). Defining "Fake News": A typology of scholarly definitions. *Digital Journalism*, 6(2), 137–153.

Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). FEVER: A large-scale dataset for fact extraction and verification. *Proceedings of NAACL 2018*.

Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. *Science*, 359(6380), 1146-1151.

Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., & Gao, J. (2018). EANN: Event adversarial neural networks for multi-modal fake news detection. *Proceedings of KDD '18*.

Zarocostas, J. (2020). How to fight an infodemic. *The Lancet*, 395(10225), 676.

Zhou, X., & Zafarani, R. (2018). Fake news: A survey of research, detection methods, and opportunities. *arXiv preprint* arXiv:1812.00315.

Bölüm 3

MAKİNE ÖĞRENMESİ VE YAPAY ZEKÂ YÖNTEMLERİNİN KUYRUK AĞLARINA UYGULANIŞI: KURAMSAL, SİMÜLASYON VE KARŞILAŞTIRMALI BİR İNCELEME

Müjgan Zobu[1]
Vedat SAĞLAM[2]

1 Doç. Dr. Amasya Üniversitesi, Eğitim Bilimleri Bölümü. ORCID: 0000-0001-9182-8927
2 Prof. Dr. Ondokuz Mayıs Üniversitesi, İstatistik Bölümü. ORCID: 0000-0002-8586-1373

1. Giriş

Gerçek hayat problemlerinde karşımıza çıkan karmaşık sistemlerin büyük bir bölümü, sınırlı kaynakların bulunduğu kuyruk/bekleme süreçleri barındırmaktadır. Telekomünikasyon ağları, bilgisayar ve veri merkezleri, üretim hatları, ulaşım sistemleri ve sağlık hizmetleri gibi pek çok alanda, ilgili sistemin performansını belirleyen temel bileşen; taleplerin/müşterilerin sisteme varış biçimi, sistemin disiplini ve bu iki sürecin ortaya çıkardığı kuyruk sistemidir. Bu tür sistemlerin modellenmesi ve analizinde Kuyruk teorisi ve Kuyruk Ağları temel bir araç olarak kullanılmaktadır (Kleinrock L., 1975), (Kelly, F. P., 1979). Klasik kuyruk teorisi matematiksel temellere dayanan stokastik modeller aracılığıyla sistem performansına ilişkin önemli sayısal sonuçlar sunmaktadır. Ancak bu modeller hemen hemen her zaman durağanlık, bağımsızlık ve belirli olasılık dağılımlarına dayalı varsayımlar altında geliştirilmiştir. Gerçek dünyada uygulamada karşımıza çıkan sistemler incelendiğinde, zamanla değişen talep/müşteri oranları, heterojen servis süreleri ve karmaşık ağ topolojileri gibi farklı nedenlerden dolayı bu varsayımlar sıklıkla sağlanamaz. Özellikle büyük ölçekli kuyruk ağlarında analitik çözümler ya elde edilemez ya da pratikte uygulanamayacak kadar karmaşıktır (Boucherie, R. J., & Van Dijk, N. M., 2017). Bu zorluklar, kuyruk ağlarının analizinde veri temelli öğrenmeye dayalı yaklaşımlara olan ilgiyi artırmıştır. Makine öğrenmesi yöntemleri, karmaşık sistemlerden elde edilen büyük hacimli veriler aracılığıyla, önceden belirlenmiş olasılık dağılımlarına ihtiyaç duymadan sistem davranışlarını öğrenebilme imkânı sunmaktadır. Son yıllarda yapılan çalışmalar, makine öğrenmesi yöntemlerinin bekleme süresi ve kuyruk uzunluğu tahmini ile sisteme ait performans ölçülerinin yaklaşık hesaplanmasında başarılı sonuçlar verdiğini göstermektedir (Gelenbe E., & Nguyen T., 2019), (Zhang Y., Chen M., & Yin Y., 2021). Öte yandan daha geniş bir çerçeve sunan yapay zekâ yaklaşımları ise yalnızca tahmin problemleriyle sınırlı kalmayıp, karar verme ve kontrol mekanizmalarını da kapsamaktadır. Özellikle pekiştirmeli öğrenme tabanlı yöntemler, kuyruk ağlarını dinamik bir kontrol problemi olarak ele alarak, yönlendirme, sunucu atama ve kaynak paylaşımı gibi kararların eş zamanlı biçimde öğrenilmesini mümkün kılmaktadır (Sutton R. S., & Barto A. G., 2018), (Mao H., Alizadeh M., Menache I., & Kandula S., 2016).

Son yıllarda derin öğrenme yöntemlerinin popüler hale gelmesi ve de hesaplama araçlarının artması ile orantılı olarak, kuyruk ağları ile yapay zekâ

arasındaki güçlü bir etkileşim ortaya çıkmıştır. Derin sinir ağları, yüksek boyutlu durum uzaylarına sahip kuyruk ağlarında performans ölçüleri tahmini ve kuyruk disiplini öğrenimi için kullanılmaya başlanmıştır (Chen, X., Wang, J., & Li, Q., 2020). Ayrıca, klasik Kuyruk Teorisi ile makine öğrenmesini birleştiren hibrit yaklaşımlar hem teorik hem de uygulamalı performans açısından olumlu sonuçlar ortaya koymaktadır (Gelenbe, E., Gellman, M., & Mang, X., 2021).

Bu kitap bölümünün amacı, kuyruk ağlarının matematiksel temelleri ile makine öğrenmesi ve yapay zekâ yaklaşımlarına bütüncül bir bakış sunmak ve bu yöntemlerin kuyruk ağlarına uygulanışını sistematik biçimde incelemektir.

2. Kuyruk Ağlarının Matematiksel Temelleri

Bu bölümde, ilk önce okuyucu için kuyruk ağlarının matematiksel altyapısı sistematik olarak özetlenerek tanıtılmaya çalışılmıştır. Bu kısmın amaca klasik kuyruk teorisinin temel kavramlarını ve varsayımlarını açıkça tanımlamak, ardından bu modellerin hangi noktalarda sınırlı kaldığını veya uygulama açısından yaşanabilecek zorlukları ortaya koymaktır.

2.1. Kuyruk Sistemleri

Çok basit haliyle, bir kuyruk sistemi, rastgele zamanlarda sisteme gelen taleplere (müşteri, paket, iş, görev vb.) sınırlı sayıda servis kaynağı tarafından sunulan hizmeti stokastik olarak modelleyen bir yapıdır. Bir kuyruk sistemi: Geliş süreci, Servis süreci, Sunucu sayısı, Kuyruk disiplini ve Sistem kapasitesi temel bileşenleriyle tanımlanır. Şimdi bu bileşenleri matematiksel olarak tanıtalım. Bu kısımdan itibaren, çalışmamızın geri kalan bölümünde sisteme gelen "talepleri", "müşteriler" olarak anacağız.

2.1.1. Geliş Süreci

Geliş süreci, müşterilerin sisteme hangi zamanlarda ulaştığını belirleyen stokastik bir süreçtir. Uygulamada kabul gören en yaygın varsayım, gelişlerin Poisson süreçli olduğudur. Yani, gelişler arası süreler λ, $(\lambda > 0)$ parametreli üstel dağılımlıdır. Poisson geliş sürecinde, sisteme gelen müşteri sayılarının olasılık dağılımı (Sağlam, V., Yücesoy, E., Sağır, M., Zobu, M., 2015),

$$P(N(t) = k) = \frac{(\lambda t)^k}{k!} e^{-\lambda t};\ \ k = 0, 1, 2, \ldots. \qquad (1)$$

şeklinde ifade edilir. Burada $N(t)$, $(0, t]$ zaman aralığında gerçekleşen geliş sayısını göstermektedir. Geliş sürecinin Poisson olması varsayımı

hesaplamalarda kolaylık sağlamakta ancak yukarıda bahsettiğimiz gerçek dünya sistemlerinde, zamanla değişen ve gelişlerin bağımlı olduğu modellerde yetersiz kalmaktadır.

2.1.2. Servis (Hizmet) Süreci

Hizmet süreci, bir müşterinin sistemde ne kadar süreyle hizmet aldığına ilişkin tesadüfi değişkenleri tanımlar. Uygulamalarda genellikle hizmet sürelerinin bağımsız ve özdeş dağılımlı oldukları kabul edilir. En basit haliyle hizmet süreleri üstel dağılımlı olup hizmet oranı μ, $(\mu > 0)$ parametresi ile gösterilir ve üstel dağılımın beklenen değeri gereği hizmet süresinin beklenen değeri (Yücesoy, E., Sağlam, V., 2016),

$$E[S] = \frac{1}{\mu} \tag{2}$$

eşitliği ile hesaplanır. Ancak gerçek sistemlerde hizmet süreleri çoğunlukla ağır kuyruklu (iş yükünün fazla olduğu) dağılımlar sergileyebilmekte ve bu durum yukarıdaki varsayımların geçerliliğini sınırlandırmaktadır.

2.1.3. Kuyruk Disiplini ve Sistem Durumu

Bir kuyruk sisteminin kuyruk disiplini, müşterilerin hangi sırayla hangi kurallara göre hizmet alacağını ifade eder. Çalışmalarda kullanılan belli başlı kuyruk disiplinleri aşağıda verilmiştir:

• İlk gelen ilk hizmet alır disiplini (FCFS).

• Son gelen ilk hizmet alır disiplini (LCFS).

• Öncelikli hizmet disiplini

• Rastgele hizmet disiplini.

Sistemin durumu ise genellikle, $X(t) = t$, t anındaki sistemdeki müşteri sayısı ile belirli bir stokastik süreç ile ifade edilir. Uygun varsayımlar altında $X(t)$, sürekli zamanlı bir Markov zinciri olarak modellenir.

2.1.4. Performans Ölçüleri

Bir kuyruk sisteminin performansı ortalama kuyruk uzunluğu $E[L]$, sistemde ortalama kalma süresi $E[W]$, kuyrukta ortalama bekleme süresi $E[W_q]$ ve sunucu kullanım oranı $\rho = \lambda/c\mu$ (c sunucu sayısını göstermektedir) ile ölçülür. Bu ölçülere *performans ölçüleri* denir. Bu ölçüler arasında *Little Yasası* olarak bilinen, (Sağlam, V. vd, 2014).

$$E[L] = \lambda E[W] \tag{3}$$

eşitliği vardır. Sistemdeki dağılımın varsayımlarından bağımsız olması nedeniyle (3) eşitliği Kuyruk Teorisinin en kullanışlı ve etkin bağıntılarından biridir.

2.2. Kuyruk Modelleri

Kuyruk sistemleri, Kendall notasyonu olarak isimlendirilen $A/S/c/K/N/D$ şeklinde bir gösterimle ifade edilirler. Bu gösterimde, A: Geliş sürecinin dağılımını, S: hizmet süresi dağılımını, c: Sunucu sayısını, K: Sistem kapasitesini, N: Geliş kaynağının büyüklüğünü ve D: Kuyruk disiplininin ifade eder. Literatürde sıklıkla, sistem kapasitesinin ve geliş kaynağı büyüklüğünün sonsuz olduğu varsayılarak bu notasyon sadeleştirilir.

2.2.1. M/M/1 Kuyruk Sistemi

En temel kuyruk modeli olan $M/M/1$ sistemi, Poisson gelişli, üstel hizmet süreli ve tek sunuculu kuyruk sistemidir. Sistem kararlı ise, sistemin trafik yoğunluğu ρ,

$$\rho = \frac{\lambda}{\mu} < 1 \tag{4}$$

koşulunu sağlamalıdır. Durağanlık varsayımı altında sistemde n tane müşteri bulunma olasılığı, (Sağlam, V., Sağır, M., Yücesoy, E., Zobu, M., 2015),

$$P_n = (1-\rho)\rho^n, \;\; n = 0,1,2,\ldots \tag{5}$$

eşitliği ile hesaplanır. Ayrıca bu kuyruk modeline ait ortalama kuyruk uzunluğu ve sistemde ortalama bekleme süresi sırasıyla,

$$E[L] = \frac{\rho}{1-\rho}\,, \;\; E[W] = \frac{1}{\mu - \lambda} \tag{6}$$

eşitlikleri ile hesaplanabilirler.

2.2.2. $M/M/c$ Kuyruk Sistemi

$M/M/c$ modeli, birden fazla paralel olarak hizmet veren sunucuların bulunduğu kuyruk sistemidir. Çalışmada önceden de belirtildiği gibi c sistemdeki sunucu sayısıdır. $M/M/c$ sistemin kararlılık koşulu,

$$\rho = \frac{\lambda}{c\mu} < 1 \tag{7}$$

şeklinde belirtilir. $M/M/c$ sisteminde performans ölçüleri *Erlang-C* formülü yardımıyla hesaplanır. Bu sistemde sunucu sayısının artmasıyla birlikte sisteme ait çeşitli hesaplamalar hızla karmaşıklaşmaktadır.

$M/M/1$ ve $M/M/c$ gibi modeller, analitik açıdan güçlü olmakla birlikte *bağımsız ve durağan gelişler*, *üstel servis süreleri* ve *basit yönlendirme ve servis disiplini* varsayımlarına sıkı şekilde bağlıdır. Bu varsayımlar ihlal edildiğinde, kapalı biçimde çözümler ya elde edilemez ya da yüksek boyutlu durum uzayları nedeniyle uygulamalarda kullanışlı değildir. Bu durum, özellikle kuyruk ağları ile modellenmiş problemlerde daha belirgin hale gelmektedir.

2.3. Kuyruk Ağları

Tek bir kuyruk sisteminden müteşekkil modeller birçok uygulama için yeterli olsa da gerçek dünya problemlerinde pek çok model birden fazla servis istasyonunun bir araya gelmesiyle oluşmaktadır. Bu tür sistemler, müşterilerin bir kuyruktan diğerine yönlendirildiği yapılara sahiptir ve kuyruk ağları olarak adlandırılır. Kuyruk ağları, çok aşamalı hizmet süreçlerini, yönlendirme politikalarını ve sistem içi geri beslemeleri modellemek için geliştirilmiş çok daha geniş bir çerçeve imkânı sunar. Matematiksel olarak bir kuyruk ağı, N adet hizmet düğümünden oluşan bir ağ yapısı olarak tanımlanır. Her düğüm, kendi geliş sürecine, hizmet mekanizmasına ve kuyruk disiplinine sahiptir. Ağ içerisindeki müşteriler, hizmetlerini aldıktan sonra sonra belirli olasılıklarla başka düğümlere yönlendirilir ya da ağdan ayrılır. Bir kuyruk ağı temel olarak; hizmet düğümü sayısı (N), $i.$ düğüme dışarıdan gelen ortalama geliş oranı (λ_i), $i.$ düğümdeki hizmet oranı (μ_i), $i.$ düğümdeki servis birimi sayısı (c_i) ve geçiş olasılıkları matrisi ($P = [p_{ij}]$) bileşenlerinden oluşur. Geçiş olasılıkları matrisinde, p_{ij} olasılığı,

$$p_{ij} = P(\text{servisi tamamlayan müşteri } i \text{ düğümünden } j \text{ düğümüne gider})$$

ve bir müşterinin sistemden çıkış olasılığı,

$$p_{i0} = 1 - \sum_{j=1}^{N} p_{ij} \tag{8}$$

eşitliği ile bulunur. Ayrıca herhangi bir düğüme gelen toplam *efektif geliş oranı* aşağıdaki gibi hesaplanır:

$$\Lambda_i = \lambda_i + \sum_{j=1}^{N} \Lambda_i p_{ji}\,; \quad i = 1, 2, \dots, N \tag{9}$$

(9) eşitliği ile verilen doğrusal denklem sistemi, ağdaki *trafik denge denklemleri* olarak adlandırılır.

2.3.1. Açık ve Kapalı Kuyruk Ağları

Kuyruk ağları, müşteri akışına göre iki temel sınıfa ayrılır:

Açık Kuyruk Ağları

- Dışarıdan ağa girişlere müsaade edilir.
- Müşteriler hizmetlerini tamamladıktan sonra ağdan ayrılabilir.
- Trafik denge denklemleri çözülebilir durumdadır.

Açık kuyruk ağları, bilgisayar ve iletişim ağlarının modellenmesinde yaygın olarak kullanılır.

Kapalı Kuyruk Ağları

- Ağa dışarıdan giriş izni yoktur.
- Ağ içindeki toplam müşteri sayısı sabittir.
- Müşteriler ağ içinde sürekli dolaşırlar.

Kapalı ağlar, zaman paylaşımının söz konusu olduğu sistemler ve kapalı üretim hatları için uygun modeller elde edilmesinde kullanışlıdır.

2.3.2. Jackson Ağları

Kuyruk ağları teorisinin en temel ve en önemli sonucu, Jackson Ağları olarak bilinen model sınıfıdır. Jackson ağları, her düğümün $M/M/1$ veya $M/M/c_i$ tipi kuyruklar olduğu açık ağlardır. Bu ağlar, dış gelişlerin Poisson süreçli, hizmet sürelerinin üstel dağılımlı, geçiş olasılıklarının sabit olduğu ve ağdaki kuyruk disiplinlerinin FCFS politikasına uyduğu varsayımları altında tanımlanırlar. Bu varsayımlar sağlandığında, ağın durağan dağılımı çarpımsal biçimde bir yapıya sahiptir.

Çarpımal-Biçim Çözümü

Jackson ağının sistem durumu, $X = (X_1, \dots, X_N)$ şeklinde tanımlansın. Burada X_i, i. düğümdeki müşteri sayısını göstermektedir. Durağan durumda,

$$P(X = n) = \prod_{i=1}^{n} (1 - \rho_i)\rho_i^{n_i} \tag{10}$$

şeklinde ayrıştırılabilir bir dağılım elde edilir. Bu eşitlikte,

$$\rho_i = \frac{\Lambda_i}{c_i \mu_i} \tag{11}$$

i. düğümün kullanım oranıdır. (10) ve (11) eşitlikleri ile verilen denklemler son derece önemlidir çünkü yüksek boyutlu bir Markov zincirinin durağan dağılımı, düğüm bazında bağımsız gibi davranmaktadır.

Jackson ağları, kuyruk teorisinin en önemli analitik kazanımları olmakla beraber; üstel dağılım varsayımı, sabit yönlendirme olasılıkları, sistem durumuna bağlı kontrol politikalarının yokluğu ve zamanla değişen trafik koşullarının modellenememesi gibi önemli sınırlamalar dolayısıyla gerçek sistemlerin tamamını kapsamakta yetersiz kalmaktadır. Bundan dolayı daha genel kuyruk ağları için kapalı biçimde çözümlerin elde edilmesi büyük ölçüde imkânsız hale gelmektedir. Özellikle durum-uzayının boyutu ağ büyüklüğüyle üstel olarak arttığından, klasik Markov analizleri pratikte uygulanamaz hale gelmektedir.

Yukarıda belirtilen matematiksel zorluklar, kuyruk ağlarının analizinde yeni yöntemlere olan ihtiyacı açıkça ortaya koymaktadır. Özellikle; analitik olarak çözülemeyen ağlar, zamanla değişen ve belirsiz ortamlar, duruma bağlı kontrol ve yönlendirme problemleri gibi durumlar, makine öğrenmesi ve yapay zekâ tabanlı yaklaşımlar için bir uygulama alanı oluşturmaktadır. Bu nedenle modern öğrenme tabanlı yöntemler Kuyruk Ağları ve Kuyruk Teorisi için yalnızca bir alternatif değil, aynı zamanda güçlü bir referans noktası olarak düşünülmelidir. Öğrenme tabanlı yöntemler, karmaşık sistem davranışlarını doğrudan veriden öğrenebilir ayrıca tahmin ve kontrol problemlerini tek bir çerçevede ele alabilirler.

Son yıllarda yapılan çalışmalar, klasik Kuyruk Teorisinin varsayımlarından doğan sınırlılıklarını aşmak amacıyla makine öğrenmesi ve yapay zekâ tabanlı yaklaşımların giderek daha fazla benimsendiğini açıkça göstermektedir. Özellikle derin öğrenme ve pekiştirmeli öğrenme yöntemleri, analitik olarak çözülemeyen veya durağanlık varsayımının geçerli olmadığı kuyruk ağlarında hem performans tahmini hem de dinamik kontrol problemleri için etkili çözümler sunmaktadır [4], [8]. Güncel araştırmalar, veri odaklı modellerin yalnızca yaklaşık sonuçlar üretmekle kalmayıp, aynı zamanda klasik teorik sonuçlarla tutarlı davranışlar sergilediğini ve hatta bazı durumlarda analitik yöntemleri geride bırakan performanslar elde ettiğini ortaya koymuştur [9], [10]. Bu eğilim, kuyruk ağları alanında analitik ve öğrenmeye dayalı yaklaşımların birlikte ele alındığı hibrit bir araştırma alanının ortaya çıktığına işaret etmektedir.

3. Makine Öğrenmesi ve Yapay Zekâ

Makine Öğrenmesi, çok basit olarak, bir sistemin açıkça programlanmadan, verilerden örüntüler öğrenerek belirli görevleri yerine getirmesini amaçlayan yöntemler bütünü olarak tanımlanabilir. Matematiksel olarak, makine öğrenmesi problemi genellikle bir girdi-çıktı ilişkisini temsil eden bilinmeyen bir fonksiyonun, gözlenen veriler yardımıyla yaklaşık olarak

öğrenilmesi şeklinde ifade edilir. Bir öğrenme problemi şu şekilde modellenebilir:

$$y = f(x) + \varepsilon \tag{12}$$

Bu eşitlikte x, sistemin gözlenen durumunu veya girdilerini; y, ilgilenilen çıktıyı (örneğin bekleme süresi veya kuyruk uzunluğu); ε ise gürültü terimini (en geniş tabiri ile hatayı) temsil etmektedir. Amaç, sınırlı gözlem verisi kullanarak $f(\cdot)$ fonksiyonunun iyi bir tahminini elde etmektir. Kuyruk ağları bağlamında bu yaklaşım, sistem performans ölçülerinin analitik modeller yerine doğrudan gözlemlerden öğrenilmesini mümkün kılmaktadır. Makine öğrenmesi yöntemleri genel olarak üç ana sınıfa ayrılmaktadır. Bunlardan ilki olan *denetimli öğrenme*, etiketli veriler ile modelin eğitilmesini esas alır. Kuyruk ağlarında denetimli öğrenme, ortalama bekleme süresi ve kuyruk uzunluğu tahmini ve sistem tıkanıklık olasılıklarının hesaplanması problemlerinde kullanılabilir. Bu uygulamalarda *regresyon*, *karar ağaçları* ve *derin sinir ağları* gibi metotlar ön plana çıkmaktadır. *Denetimsiz öğrenme* ise etiketli çıktılar olmaksızın, verideki gizli yapıları ve örüntüleri keşfetmeyi amaçlar. Kuyruk ağlarında denetimsiz öğrenme; trafik profillerinin sınıflandırılması ve benzer sistem durumlarının gruplanması gibi problemler için kullanılır.

3.1. Yapay Zekâ ve Karar Verme

Makine öğrenmesi, yapay zekânın önemli bir alt alanı olmakla birlikte, Yapay Zekâ daha geniş bir çerçeve sunar. Yapay zekâ, yalnızca tahmin yapmayı değil, aynı zamanda akıl yürütme, karar verme ve hedef odaklı davranış üretmeyi de kapsar. Kuyruk ağları açısından bakıldığında, temel problem yalnızca sistem performansının tahmin edilmesi değil, bu performansı iyileştirecek kararların alınmasıdır. Bu kararlar; yönlendirme, sunucu atama, önceliklendirme ve kaynak paylaşımı gibi kontrol mekanizmalarını kapsar. Yapay zekâ yöntemleri, bu tür karar problemlerini bütüncül bir çerçevede ele alma imkânı sunmaktadır. Makine öğrenmesi yöntemleri arasında, kuyruk ağları ile en güçlü kavramsal uyuma sahip yaklaşım *Pekiştirmeli Öğrenmedir*. Pekiştirmeli öğrenme, bir öğrenici ile çevre arasındaki etkileşim üzerine kuruludur ve öğrenici, aldığı ödüller doğrultusunda en iyi davranış politikasını öğrenmeye çalışır. Bu çerçevede bir kuyruk ağı şu şekilde modellenebilir:

• Durum: Kuyruk uzunlukları, sunucu dolulukları

• Eylem: Yönlendirme kararı, servis ataması

• Ödül: Negatif bekleme süresi, sistem maliyeti

• Politika: Durumdan eyleme geçiş kuralı

Bu yapı, kuyruk ağlarını doğal bir biçimde Markov Karar Süreci (MDP) olarak ele almayı mümkün kılar. Analitik yöntemlerle çözülmesi zor olan kontrol problemleri, bu sayede deneyim yoluyla öğrenilebilmektedir.

3.2. Derin Öğrenme ve Yüksek Boyutlu Sistemler

Kuyruk ağları büyüdükçe, durum uzayının boyutu hızla artmakta ve klasik öğrenme yöntemleri yetersiz kalmaktadır. Bu noktada derin öğrenme teknikleri devreye girmektedir. Derin sinir ağları, yüksek boyutlu durum uzaylarını kompakt temsillere dönüştürerek, karmaşık karar politikalarının öğrenilmesini mümkün kılmaktadır.

Derin pekiştirmeli öğrenme yaklaşımları, özellikle büyük ölçekli ağlarda, klasik optimizasyon yöntemlerine kıyasla daha esnek ve ölçeklenebilir çözümler sunmaktadır. Bu durum, modern kuyruk ağlarının analizinde derin öğrenmenin neden giderek daha fazla tercih edildiğini açıklamaktadır.

Makine öğrenmesi ve yapay zekâ yöntemleri, klasik kuyruk teorisinin yerine geçmekten ziyade, onu tamamlayan araçlar olarak değerlendirilmelidir. Analitik modeller, sistem davranışına ilişkin önemli yapısal içgörüler sunarken; öğrenme tabanlı yöntemler, bu içgörülerin yetersiz kaldığı karmaşık ve durağan olmayan ortamlarda devreye girer. Bu nedenle, son yıllarda analitik kuyruk teorisi ile makine öğrenmesini birleştiren hibrit yaklaşımlar hem teorik sağlamlık hem de uygulama performans açısından ön plana çıkmaktadır.

4. Makine Öğrenmesinin Kuyruk Ağlarına Uygulanışı

Bu bölümde, Makine Öğrenmesi yöntemlerinin kuyruk ağlarına uygulanışı üzerine bir bakış sunulacaktır.

4.1. Performans Tahmini Problemleri

Makine öğrenmesinin kuyruk ağlarında en yaygın kullanım alanlarından biri, sistem performans ölçülerinin tahmin edilmesidir. Bekleme süresi, kuyruk uzunluğu, gecikme dağılımları ve kayıp olasılıkları gibi büyüklükler, klasik modellerde çoğu zaman kapalı formda elde edilememektedir. Bu tür durumlarda, öğrenme tabanlı yaklaşımlar doğrudan gözlem verileri üzerinden tahmin üretmektedir. Matematiksel olarak, performans tahmini problemi şu şekilde ifade edilebilir:

$$\hat{y} = f_\theta(x) \tag{13}$$

Burada x, sistem durumunu veya parametrelerini (geliş oranları, hizmet kapasiteleri, yönlendirme oranları vb. $\hat{y}$ ise tahmin edilen performans

ölçütünü temsil etmektedir. $f_\theta(\cdot)$, öğrenme algoritması tarafından θ parametreleri üzerinden belirlenen bir modeldir. Bu çerçevede doğrusal regresyon, rastgele ormanlar ve derin sinir ağları gibi yöntemler kullanılarak, karmaşık kuyruk ağlarının yaklaşık performans modelleri başarıyla inşa edilebilmektedir.

4.2. Parametre Öğrenimi ve Model Kalibrasyonu

Klasik kuyruk modelleri, geliş ve hizmet süreçlerine ilişkin parametrelerin önceden bilindiğini varsayar. Ancak pratikte bu parametreler çoğu zaman bilinmemekte veya zamanla değişmektedir. Makine öğrenmesi, bu parametrelerin doğrudan veriden tahmin edilmesini mümkün kılmaktadır. Örneğin, hizmet sürelerinin üstel dağılımlı olmadığı bir sistemde, hizmet süresi dağılımının momentleri veya etkin hizmet oranları veri üzerinden öğrenilebilir. Benzer biçimde, geliş süreçlerinin durağan olmadığı durumlarda, zamanla değişen geliş oranları makine öğrenmesi tabanlı zaman serisi modelleri yardımıyla tahmin edilebilmektedir.

4.3. Sistemin Trafik ve Müşteri Tahmini

Kuyruk ağlarının performansı büyük ölçüde müşteri ve trafik (iş yükü) yoğunluğuna. Bu nedenle, gelecekteki geliş oranlarının doğru biçimde tahmini, sistem planlaması ve kapasite yönetimi açısından kritik öneme sahiptir. Makine öğrenmesi tabanlı zaman serisi modelleri, geçmiş trafik gözlemlerinden yararlanarak gelecekteki müşteri seviyelerini tahmin edebilmektedir. Bu bağlamda, tekrarlayan sinir ağları ve uzun-kısa süreli bellek (LSTM) yapıları, zamansal bağımlılıkların güçlü olduğu kuyruk sistemlerinde etkili sonuçlar vermektedir. Bu tür tahminler, klasik kuyruk teorisinde genellikle sabit kabul edilen paramet**relerin dinamik biçimde güncellenmesini mümkün kılmaktadır.**

4.4. Simülasyon Destekli Öğrenme Yaklaşımları

Makine öğrenmesi, simülasyon verileriyle eğitilerek, pahalı simülasyon çalıştırmalarının yerini alabilecek hızlı modeller üretme potansiyeline sahiptir. Bu yaklaşımda, simülasyon ortamı bir veri üretici olarak kullanılır ve öğrenilen model, farklı sistem konfigürasyonları için performans tahminleri yapar. Böylece, parametre uzayının geniş olduğu durumlarda bile, hesaplama maliyeti önemli ölçüde azaltılabilir.

4.5. Genelleme ve Modelin Güvenilirliği

Makine öğrenmesi tabanlı kuyruk ağları modellerinin en kritik yönlerinden biri, öğrenilen modellerin genelleme yeteneğidir. Eğitim verisiyle sınırlı

kalan modeller, daha önce gözlemlenmemiş sistem durumlarında hatalı tahminler üretebilir. Bu nedenle, kuyruk teorisinden elde edilen yapısal bilgiler (örneğin kararlılık koşulları, Little Kanunu gibi temel ilişkiler), öğrenme sürecine kısıt olarak entegre edilmektedir. Bu tür öğrenme yaklaşımları, modelin güvenilirliğini ve yorumlanabilirliğini artırmaktadır.

Bu bölümde ele alınan yöntemler, makine öğrenmesinin kuyruk ağlarında özellikle tahmin, parametre öğrenimi ve performans modelleme problemlerinde güçlü araçlar sunduğunu göstermektedir. Bununla birlikte, bu yaklaşımlar genellikle pasiftir; yani sistem davranışını tahmin eder ancak doğrudan kontrol etmeyi hedeflemez. Bir sonraki bölümde, bu sınırlamanın ötesine geçilerek, kuyruk ağlarının aktif biçimde kontrol edilmesini amaçlayan yapay zekâ ve pekiştirmeli öğrenme tabanlı yaklaşımlar üzerinde durulacaktır.

5. Yapay Zekâ Ve Pekiştirmeli Öğrenme Tabanlı Yaklaşımlar

Önceki bölümde ele alınan makine öğrenmesi yöntemleri, kuyruk ağlarının davranışlarını tahmin etme ve modelleme açısından önemli avantajlar sunmaktadır. Ancak modern sistemlerde temel problem çoğu zaman yalnızca performansın tahmin edilmesi değil, aynı zamanda sistemin aktif olarak kontrol edilmesidir. Bu bağlamda yapay zekâ, özellikle Pekiştirmeli Öğrenme, kuyruk ağlarını dinamik karar verme problemleri olarak ele alarak klasik yaklaşımların ötesine geçmektedir.

5.1. Bir Kontrol Problemi Olarak Kuyruk Ağları

Kuyruk ağlarında kontrol problemleri genellikle aşağıdaki kararların alınıp alınamayacağını tahmin etmeye çalışır:

• müşterilerin hangi düğüme yönlendirileceği

• hizmet birimlerinin hangi kuyruklara tahsis edileceği

• önceliklendirme ve hizmet sıralamasının belirlenmesi

• kaynakların zamanla yeniden dağıtılması

Bu kararlar, sistemin anlık durumuna bağlı olarak verilmek zorundadır ve yanlış kararlar bekleme sürelerinde ciddi artışlara yol açabilmektedir.

5.2. Markov Karar Süreci Formülasyonu

Pekiştirmeli öğrenme çerçevesinde bir kuyruk ağı, genellikle bir Markov Karar Süreci (MDP) olarak modellenir. Bu model; Durum uzayı (S): Kuyruk uzunlukları, hizmet birimi dolulukları, sistem yükü; Eylem uzayı ($\mathcal{A}$): Yönlendirme kararları, hizmet atamaları; Geçiş olasılıkları: Sistem

dinamikleri; Ödül fonksiyonu: Bekleme süresi, gecikme veya maliyetin negatifi, bileşenlerden oluşur.

Amaç, beklenen uzun dönem ödülü maksimize eden bir $\pi(a|s)$ politikası öğrenmektir:

$$\max_{\pi}\left[E\sum_{t=0}^{\infty}\Upsilon^{t}r_{t}\right] \tag{14}$$

Bu eşitlikte $\Upsilon \in (0,1)$ iskonto katsayısını temsil etmektedir. Bu yaklaşım, kuyruk ağlarında optimal kontrol politikalarının analitik olarak elde edilemediği durumlar için güçlü bir alternatif sunmaktadır.

5.3. Modelden-Bağımsız Öğrenme Yaklaşımları

Birçok uygulamada kuyruk sisteminin geçiş olasılıkları bilinmemektedir. Bu durumda modelden-bağımsız pekiştirmeli öğrenme yöntemleri ön plana çıkmaktadır. Bu yöntemler, sistem dinamiklerini açıkça modellemeden, doğrudan etkileşim yoluyla öğrenir. Özellikle büyük ve karmaşık kuyruk ağlarında, deneyim yoluyla öğrenilen bu politikalar; sabit veya sezgisel yönlendirme kurallarına kıyasla daha düşük bekleme süreleri ve daha dengeli kaynak kullanımı sağlayabilmektedir. Bu durum, yapay zekâ tabanlı yöntemlerin pratik üstünlüğünü açıkça ortaya koymaktadır.

5.4. Derin Pekiştirmeli Öğrenme ve Ölçeklenebilirlik

Kuyruk ağlarının düğüm sayısı arttıkça, durum uzayının boyutu hızla büyümektedir. Bu durum, klasik tablo-temelli pekiştirmeli öğrenme yöntemlerini uygulanamaz hale getirir. Derin pekiştirmeli öğrenme yaklaşımları, bu sorunu derin sinir ağları aracılığıyla aşmayı hedefler. Derin ağlar, yüksek boyutlu durumları daha düşük boyutlu temsillere dönüştürerek karmaşık politikaların öğrenilmesini mümkün kılar. Bu sayede, büyük ölçekli kuyruk ağlarında bile çevrim içi ve uyarlanabilir kontrol stratejileri geliştirilebilmektedir.

5.5. Kararlılık ve Öğrenim Dengesi

Pekiştirmeli öğrenmenin kuyruk ağlarına uygulanmasındaki en kritik zorluklardan biri, öğrenme süreci ile sistem kararlılığı arasındaki dengedir. Aşırı öğrenme sistem performansını geçici olarak bozabilirken; yetersiz öğrenme, alt-optimal politikalara sıkışılmasına yol açabilmektedir. Bu nedenle, kuyruk teorisinden elde edilen kararlılık koşulları ve yapısal bilgiler, pekiştirmeli öğrenme algoritmalarına rehber olarak entegre

edilmektedir. Bu tür yaklaşımlar hem öğrenme sürecini hızlandırmakta hem de sistemin güvenli çalışmasını sağlamaktadır.

5.6. Yapay Zekâ Tabanlı Kontrolün Değerlendirilmesi

Yapay zekâ ve pekiştirmeli öğrenme tabanlı yöntemler, kuyruk ağlarında aşağıdaki açılardan önemli avantajlar sunmaktadır:

• Analitik olarak çözülemeyen kontrol problemlerinin ele alınabilmesi

• Zamanla değişen ve belirsiz ortamlara uyum

• Büyük ölçekli sistemlerde ölçeklenebilirlik

• Klasik politikalara kıyasla daha iyi uzun dönem performansı

Bununla birlikte, öğrenme süresince oluşabilecek performans kayıpları, yorumlanabilirlik eksikliği ve teorik sınırlamalar, bu yaklaşımların dikkatle değerlendirilmesini gerekli kılmaktadır.

6. Simülasyon Tabanlı Karşılaştırmalı Bir Çalışma

Bu bölümde, makine öğrenmesi ve yapay zekâ tabanlı yaklaşımların kuyruk ağlarındaki performansı, geleneksel kuyruk teorisi ile karşılaştırmalı olarak incelenmektedir. Amaç, önceki bölümlerde sunulan kuramsal çerçevenin, kontrollü bir simülasyon ortamında sayısal olarak değerlendirilmesidir. Gerçek sistem verilerine erişimin sınırlı olabileceği göz önünde bulundurularak, çalışma rastgele üretilmiş bir veri seti üzerinden gerçekleştirilmiştir. İncelenen sistem, tek sunuculu bir kuyruk modeli olarak ele alınmıştır, yani aşağıdaki özelliklere sahiptir:

- Kuyruk tipi: $M/M/1$
- Geliş süreci: Poisson (oran λ)
- Servis süreci: Üstel dağılım (oran μ)
- Kararlılık koşulu: $\lambda < \mu$

Bu simülasyon uygulamasında sistem, ayrık zamanlı bir simülasyon ile modellenmiştir. Her zaman adımında:

1. Yeni müşteri gelişleri rastgele üretilir
2. Kuyruğa giren müşteriler güncellenir
3. Hizmet birimi boş ise hizmet verilmeye başlanır
4. Hizmeti tamamlanan müşteri sistemden ayrılır

Simülasyon sonucunda aşağıdaki değişkenler kaydedilmiştir:

Tablo 1. *Simülasyon sonucu elde edilen değişkenler*

Değişken	Açıklama
t	Zaman adımı
Q_t	Kuyruk uzunluğu
W_t	Ortalama bekleme süresi
S_t	Sunucu durumu (boş/dolu)
λ_t	Anlık geliş oranı

Bu veri seti, hem tahmin (ML) hem de kontrol (YZ) yaklaşımlarının uygulanmasına olanak tanımaktadır.

6.1. Karşılaştırılan Yöntemler

$M/M/1$ kuyruğu için ortalama bekleme süresi analitik olarak:

$$E[W] = \frac{1}{\mu - \lambda} \tag{15}$$

şeklinde hesaplanmaktadır. Bu değer, yaptığımız simülasyon sonuçlarının karşılaştırılması için teorik referans olarak kullanılmıştır.

6.2. Makine Öğrenmesi Tabanlı Yaklaşım (Tahmin Odaklı)

Bu yöntemde problem, bir regresyon problemi olarak ele alınmıştır. Girdi değişkenleri olarak, $(Q_{t-1}, Q_{t-2}, \lambda_t)$ ve çıktı değişkeni olarak $\widehat{W}_t$ alınmıştır. Yani bu yaklaşımdaki amaç, geçmiş sistem gözlemlerine dayanarak bekleme süresinin tahmin edilmesidir. Bu yaklaşım, sistem davranışını başarılı biçimde tahmin edebilmekte; ancak doğrudan kontrol sağlamamaktadır. Makine Öğrenmesi tabanlı tahmin odaklı yaklaşım için yapılan simülasyonda kullanılan Pseudo kod aşağıda verilmiştir.

```
For each time step t:
    Observe queue length Q_t
    Collect feature vector X_t
    Predict waiting time W_hat_t = ML_Model(X_t)
```

6.3. Yapay Zekâ / Pekiştirmeli Öğrenme Tabanlı Yaklaşım (Kontrol Odaklı)

Bu yöntemde kuyruk sistemi bir karar verme problemi olarak modellenmiştir:

- Durum: Kuyruk uzunluğu Q_t
- Aksiyon: Servis hızının ayarlanması

- Ödül: $r_t = -W_t$

Amaç, uzun dönem ortalama bekleme süresini minimize eden bir politika öğretmektir. Aşağıda bu yapay zekâ tabanlı pekiştirmeli yönteme ait simülasyon çalışmasının Pseudo kod verilmiştir:

```
Initialize policy π randomly
For each episode:
    Observe current state Q_t
    Select action a_t according to π
    Apply action to system
    Observe reward r_t and next state Q_{t+1}
    Update policy π
```

Bu yöntem, sistem performansını aktif olarak iyileştirmeyi hedeflemektedir.

Karşılaştırma aşağıdaki ölçütler üzerinden yapılmıştır:

• Ortalama bekleme süresi

• Ortalama kuyruk uzunluğu

• Zaman içindeki dalgalanma (varyans)

Bu ölçütler hem sistem verimliliğini hem de kararlılığı değerlendirmek için kullanılmıştır.

7. Sayısal Sonuçlar ve Karşılaştırmalı Analiz

Bu bölümde, $M/M/1$ kuyruk sistemi için gerçekleştirilen simülasyon çalışmasından rastgele üretilen veri seti kullanılarak, üç farklı yaklaşımın performansı karşılaştırılmıştır. Karşılaştırma, farklı sistem yükleri altında ortalama bekleme süresi ve kuyruk uzunluğu ölçütleri üzerinden yapılmıştır. Simülasyonlar, üç farklı trafik yoğunluğu seviyesi için gerçekleştirilmiştir çünkü bu senaryolar, kuyruk sistemlerinin pratikte karşılaştığı tipik çalışma koşullarını temsil etmektedir.

- düşük ağ yükü: $\rho = \lambda/\mu = 0.5$
- orta ağ yükü: $\rho = \lambda/\mu = 0.8$
- ağır ağ yükü: $\rho = \lambda/\mu = 0.95$

Aşağıdaki tabloda, farklı yöntemler altında elde edilen ortalama bekleme süresi değerleri sunulmuştur:

Tablo 2. *Farklı yöntemler altında elde edilen ortalama bekleme süreleri*

Yük Seviyesi	Geleneksel Kuyruk Teorisi	Makine Öğrenmesi	Yapay Zeka/PÖ
Düşük yük (p=0.5)	2.0	2.1	2.0
Orta yük (p=0.8)	5.0	4.6	4.1
Yüksek yük (p=0.95)	20.0	17.8	13.5

Tablodan elde edilen sonuçlar aşağıdaki temel gözlemleri ortaya koymaktadır:

• Düşük yük koşullarında, tüm yöntemler benzer performans sergilemektedir. Bu durum, klasik kuyruk teorisinin durağan ve düşük yoğunluklu sistemlerde hâlen güçlü bir yöntem olduğunu göstermektedir.

• Orta yük seviyesinde, makine öğrenmesi tabanlı yaklaşımın geleneksel kuyruk teorisine kıyasla daha düşük bekleme süreleri tahmin edebildiği gözlemlenmiştir. Bu durum, makine öğrenmesi tabanlı modellerin doğrusal olmayan sistem davranışlarını daha iyi tahmin edebildiğini göstermektedir.

• Yüksek yük altında, yapay zekâ ve pekiştirmeli öğrenme tabanlı yaklaşım belirgin biçimde geleneksel yöntemden üstündür. Öğrenilen kontrol politikaları, sistemin aşırı yüklenmesini kısmen engelleyerek bekleme sürelerini önemli ölçüde azaltmıştır.

Bu sonuçlar, yapay zekâ tabanlı yöntemlerin özellikle kritik ve tıkanmaya yakın çalışma rejimlerinde klasik yaklaşımlara göre daha etkili olduğunu göstermektedir. Benzer eğilimler, ortalama kuyruk uzunluğu ölçüsü için de gözlemlenmiştir. Yapay zekâ tabanlı yaklaşım, yüksek yük koşullarında kuyruk uzunluğunu daha dengeli tutarak sistem kararlılığını artırmıştır. Makine öğrenmesi tabanlı yöntem ise tahmin doğruluğu açısından başarılı olmakla birlikte, doğrudan kontrol sağlamadığı için kuyruk büyümesini sınırlı ölçüde azaltabilmiştir. Bu karşılaştırmalı çalışma, üç temel sonucu açıkça ortaya koymaktadır:

1. Geleneksel kuyruk teorisi, teorik analiz ve referans değerler açısından vazgeçilmezdir.

2. Makine öğrenmesi, kuyruk performansının tahmin edilmesinde güçlü bir araçtır.

3. Yapay zekâ ve pekiştirmeli öğrenme, dinamik ve yüksek yük altındaki sistemlerde aktif kontrol avantajı sağlamaktadır.

Dolayısıyla, bu yöntemler rekabet eden değil, birbirini tamamlayan yaklaşımlar olarak değerlendirilmelidir.

Kaynakça

Kleinrock, L. (1975). Queueing systems, volume I: Theory. Wiley.

Kelly, F. P. (1979). Reversibility and stochastic networks. Wiley.

Boucherie, R. J., & van Dijk, N. M. (2017). Queueing networks: A fundamental approach. Springer. https://doi.org/10.1007/978-3-319-28091-9

Gelenbe, E., & Nguyen, T. (2019). Machine learning for queueing systems. ACM SIGMETRICS Performance Evaluation Review, 47(2), 36–40. https://doi.org/10.1145/3377148.3377156

Zhang, Y., Chen, M., & Yin, Y. (2021). Learning-based performance modeling of queueing networks. Performance Evaluation, 148, 102199. https://doi.org/10.1016/j.peva.2021.102199

[Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016). Resource management with deep reinforcement learning. Proceedings of the 15th ACM Workshop on Hot Topics in Networks, 50–56. https://doi.org/10.1145/3005745.3005750

Chen, X., Wang, J., & Li, Q. (2020). Deep learning for large-scale queueing systems: Modeling and control. IEEE Transactions on Network Science and Engineering, 7(4), 2452–2465. https://doi.org/10.1109/TNSE.2020.2982567

Gelenbe, E., Gellman, M., & Mang, X. (2021). Deep learning and queueing theory. Applied Soft Computing, 109, 107506. https://doi.org/10.1016/j.asoc.2021.107506

Sağlam, V.,Yücesoy, E., Sağır, M., Zobu, M. A Study on a Tandem Stochastic Queueing Model with Parallel Phases and a Numerical Example. Science Journal of Applied Mathematics and Statistics. Vol. 3, No. 2, 2015, pp. 33-38. doi: 10.11648/j.sjams.20150302.12

Yücesoy, E., & Sağlam, V. (2021). Analysis and Simulation of a Two-Stage Blocked Tandem Queueing System. Journal of New Theory(35), 91-102. https://doi.org/10.53570/jnt.938304

Sağlam, V., Uğurlu, M., Yücesoy, E., Zobu, M., Sağır, M. (2014). On Optimization of a Coxian Queueing Model with Two Phases. Applied and Computational Mathematics, 3(2), 43-47. https://doi.org/10.11648/j.acm.20140302.11

Sağlam, V., Sağır, M., Yücesoy, E., Zobu, M. (2015). The Analysis, Optimization, and Simulation of a Two-Stage Tandem Queueing Model with Hyperexponential Service Time at Second Stage, Mathematical Problems in Engineering, 2015, 165219, 6 pages, 2015. https://doi.org/10.1155/2015/165219

Bölüm 4

MAKİNE ÖĞRENMESİ İLE TIBBİ TANIDA KARAR ŞEFFAFLIĞI VE MODEL KARMAŞIKLIĞI: UCI TİROİT VERİSİ ÜZERİNDE KARŞILAŞTIRMALI BİR ANALİZ

Yunus Emre Ceylan[1]
Eralp Dogu[2]

1 Milli Savunma Üniversitesi Türkiye, yunusemre.ceylan@msu.edu.tr , https://orcid.org/0000-0003-0441-4748
2 Mugla Sitki Kocman University Department of Statistics Muğla Türkiye, eralp.dogu@mu.edu.tr, bhttps://orcid.org/0000-0002-8256-7304

1. GİRİŞ

Günümüzde sağlık bilimlerinde veri miktarının katlanarak artması, hastalıkların teşhis ve tedavi süreçlerinde bilgisayar destekli sistemlerin kullanımını bir zorunluluk haline getirmiştir. Tıbbi tanı süreçlerinde makine öğrenmesi algoritmaları, özellikle biyokimyasal verilerdeki karmaşık örüntüleri tanıma ve klinik karar destek sistemlerine temel oluşturma noktasında yüksek başarı sergilemektedir. Tiroit hastalıkları gibi hormon seviyelerindeki hassas değişimlerin takip edildiği alanlarda, algoritmaların sunduğu tahmin gücü, erken teşhis ve kişiselleştirilmiş tedavi protokollerinin geliştirilmesi açısından hayati bir öneme sahiptir. Ancak tıp literatüründe, modellerin sadece yüksek doğruluk oranına sahip olması yeterli görülmemekte; teşhisin hangi klinik parametrelere dayanılarak konulduğunun anlaşılabilir olması, yani "açıklanabilirlik" (explainability) kavramı, hasta güvenliği ve etik sorumluluk açısından ön plana çıkmaktadır.

Akademik literatürde tiroit bozukluklarının tespiti üzerine yapılan çalışmalar incelendiğinde, makine öğrenmesi yaklaşımlarının, özellikle de topluluk öğrenme ve karar ağacı temelli modellerin teşhis başarısında öne çıktığı görülmektedir. Begum ve Parkavi (2019), UCI veri havuzundan temin ettikleri 15 farklı özniteliğe sahip kayıtlar üzerinde ID3 ve C4.5 algoritmalarını test etmişlerdir. Araştırmacılar, hipertiroit ve hipotiroit tanılarında T3, T4 ve TSH gibi temel hormon seviyeleri arasındaki korelasyonun sınıflandırma başarısındaki kritik rolüne dikkat çekmişlerdir.

Veri seti çeşitliliğinin model performansı üzerindeki etkisini araştıran Kousarrizi ve arkadaşları (2012), hem UCI veri setini hem de bir hastanenin endokrinoloji servisinden toplanan 1538 vakalık kohortu analiz etmişlerdir. Destek Vektör Makineleri (SVM) yönteminin kullanıldığı çalışmada, klinik semptomların (çarpıntı, titreme, ödem vb.) biyokimyasal bulgularla birleştirilmesinin tanısal doğruluğu artırdığı vurgulanmıştır. Benzer şekilde, Akgül v.d. (2020), hipotiroit teşhisinde invaziv testlerin komplikasyon riskini azaltmak ve karar sürecini optimize etmek amacıyla 3163 örnekten oluşan geniş bir veri setiyle çalışmışlardır. Bu çalışmada, sınıflar arasındaki sayısal dengesizliği yönetmek adına çeşitli örnekleme stratejilerine başvurulmuş; Lojistik Regresyon, k-NN ve SVM modelleri aracılığıyla tanı isabetinin artırılabileceği gösterilmiştir.

Karar mekanizmalarının matematiksel temellerine odaklanan Margret v.d. (2012), UCI veri setindeki 21 nitelik üzerinden farklı bölme kurallarının (splitting rules) etkinliğini kıyaslamışlardır. Çalışma sonucunda, normalleştirilmiş bölme kriterlerinin duyarlılık ve kesinlik metriklerinde daha üstün sonuçlar verdiği ve bu yaklaşımın farklı medikal veri setlerine de genelle-

nebileceği saptanmıştır. Son olarak Banu (2017) tarafından gerçekleştirilen kapsamlı analizde, 3772 vakalık bir veri kümesi WEKA platformu üzerinden değerlendirilmiştir. Karar ağacı ve veri madenciliği tekniklerinin kıyaslandığı araştırmada; k-NN (%96,35) ve SVM (%94,44) yöntemlerine oranla, C4.5 ve Rastgele Orman algoritmalarının %99,47'lik bir doğruluk düzeyi ile tiroit sınıflandırmasında en başarılı sonuçları sunduğu rapor edilmiştir.

Makine öğrenmesi modelleri, şeffaflık düzeylerine göre geniş bir spektrumda yer almaktadır. Karar ağaçları (CART, C5.0, CTREE) (Quinlan, 1993) gibi yöntemler, insan zihninin takip edebileceği mantıksal kurallar ve hiyerarşik yapılar sunarken; Rastgele Orman (Random Forest) (Breiman, 2001) ve XGBOOST (Chen, 2016) gibi topluluk öğrenme (ensemble) yöntemleri, binlerce işlemin birleşimi sonucu oluşan "kara kutu" (black-box) modeller olarak tanımlanmaktadır. Topluluk modelleri genellikle daha yüksek tahmin başarısı sergilese de, oluşturdukları girift karar sınırları klinik uzmanların bu kararları rasyonalize etmesini zorlaştırmaktadır. Özellikle tıbbi kılavuzların (guidelines) sunduğu net eşik değerleri ile algoritmik kararlar arasındaki geometrik uyum, yapay zekanın klinik ortamlarda kabul görmesi için en kritik bariyerlerden birini oluşturmaktadır.

Bu çalışmanın temel amacı, UCI Machine Learning Repository'den alınan tiroit veri seti üzerinden ağaç temelli makine öğrenmesi algoritmalarının ürettiği karar kurallarını görselleştirerek, modellerin karmaşıklığı ile klinik açıklanabilirliği arasındaki ilişkiyi analiz etmektir. Çalışma kapsamında, tekil ağaç yapılarının sunduğu şeffaf karar geometrisi ile topluluk modellerinin yarattığı karmaşık karar sınırları karşılaştırmalı olarak incelenmiştir. Bu analizle, tıbbi tanı süreçlerinde sadece istatistiksel doğruluk metriklerinin değil, aynı zamanda karar sınırlarının tıbbi kılavuzlarla olan görsel uyumunun da bir başarı kriteri olarak değerlendirilmesi gerektiği vurgulanmaktadır. Elde edilen bulguların, hekim-bilgisayar etkileşimini güçlendirecek daha güvenilir ve denetlenebilir klinik karar destek sistemlerinin tasarımına katkı sağlaması hedeflenmektedir.

2. MATERYAL VE YÖNTEM

2.1. Veri Setinin Tanımı ve Ön İşleme

UCI Machine Learning Repository üzerinden elde edilen veri seti, toplam 215 kişiye ait klinik tiroit kayıtlarını barındırmaktadır. Bu açık veri seti, açık kaynak kodlu R programlama diline aktarılarak analiz edilmiş ve içerisinde barındırdığı üç sınıflı yanıt değişkeni (tiroit, hipotiroit ve hipertiroit) "sınıflar" başlığı altında tanımlanmıştır. Veri seti; tiroit (150), hipotiroit (35) ve hipertiroit (30) sınıflarından oluşan 3 sınıflı bir yanıt değişkenine sahiptir. Mode-

lin kurulumunda açıklayıcı değişkenler olarak; tiroit uyarıcı hormonu (TSH), triiyodotironin hormonu (sT3), tiroksin hormonu (sT4), tiroit fonksiyon testi (DTSH) ve RT3U kullanılmıştır. Çalışma kapsamında kurulan istatistiksel model, Y ~ RT3U + T4 + T3 + TSH +DTSH formülü ile ifade edilerek ilgili özniteliklerin hastalık sınıfları üzerindeki etkisini incelemeyi amaçlamaktadır. Veri seti UCI tarafından yayınlanmıştır (Quinlan, 1986).

2.2. Uygulanan Algoritmalar ve Performans Metrikleri

Tiroit hastalığının teşhisi ve sınıflandırılması amacıyla bu çalışmada, açıklanabilirlik düzeyleri farklılık gösteren beş temel ağaç temelli makine öğrenmesi algoritması kullanılmıştır. Şeffaf ve yorumlanabilir modeller kategorisinde; Gini indeksini temel alan CART (Sınıflandırma ve Regresyon Ağaçları), bilgi kazancı ve entropi ölçütlerini kullanan C5.0 ve istatistiksel anlamlılık testlerine dayalı yansız seçim yapan CTREE (Koşullu Çıkarım Ağaçları) yöntemleri tercih edilmiştir. Daha yüksek tahmin gücü hedefleyen karmaşık topluluk modelleri tarafında ise, birden fazla ağacın kombinasyonuyla çalışan Rastgele Orman (Random Forest) ve gradyan arttırma tekniğinin yüksek performanslı bir versiyonu olan XGBOOST (Aşırı Gradyan Arttırma) algoritmaları uygulanmıştır. Bu algoritmik çeşitlilik, hem klinik kılavuzlarla uyumlu basit karar yapılarını hem de verideki gizli örüntüleri yakalayan yüksek performanslı tahminleri karşılaştırmalı olarak analiz etme olanağı sunmuştur.

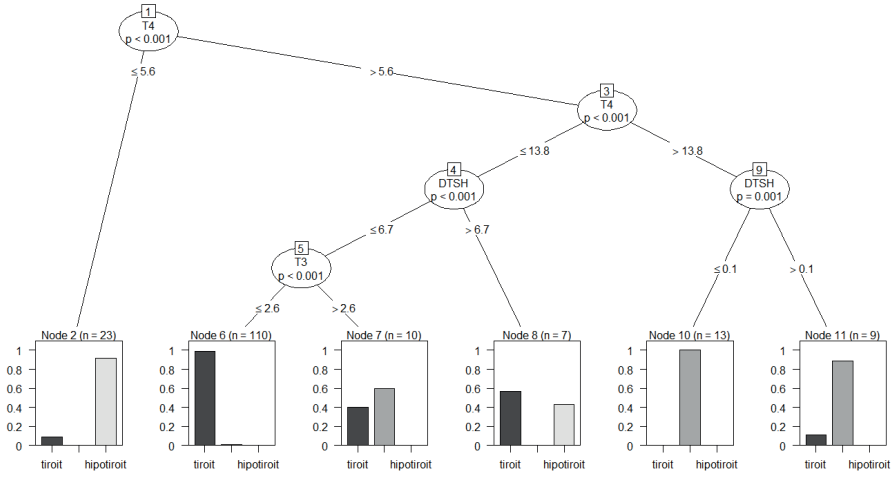
Modelin performansını objektif bir şekilde ölçmek amacıyla veri seti; %80 eğitim ve %20 test verisi olarak ikiye ayrılmıştır. Eğitim aşamasında, modellerin geçerliliğini artırmak ve aşırı uyumu (overfitting) önlemek için "fitControl" parametresiyle yapılandırılmış 10 katlı çapraz doğrulama (10-fold cross-validation) yöntemi kullanılmıştır. Bu süreçte, modellerin en uygun hiper-parametreleri (örneğin; rastgele orman için mtry, XGBOOST için eta ve gamma) doğruluk ve kappa değerleri maksimize edilecek şekilde belirlenmiştir. Sınıflandırma yöntemlerinin etkinliğini ve klinik güvenilirliğini değerlendirmek amacıyla karmaşıklık matrisi (confusion matrix) üzerinden türetilen çeşitli performans indikatörlerinden faydalanılmıştır. Modellerin genel başarısı doğruluk (accuracy) ve şans faktöründen arındırılmış uyumu gösteren kappa istatistiği ile ölçülürken , sınıfların doğru tanımlanma oranlarını belirlemek için hassasiyet (sensitivity) ve özgüllük (specificity) metrikleri kullanılmıştır. Ayrıca, modellerin kesinlik (precision) ve F skoru değerleri hesaplanarak dengeli bir performans analizi gerçekleştirilmiş ve doğruluk değerleri için %95 güven aralıkları tanımlanmıştır. Bu metrikler, özellikle modelin klinik bir rehber olarak kullanılabilirliğini belirleyen "karar noktaları" ve "karar kuralları" ile birlikte sentezlenerek, en uygun tanı yönteminin seçilmesine temel oluşturmuştur.

3. BULGULAR

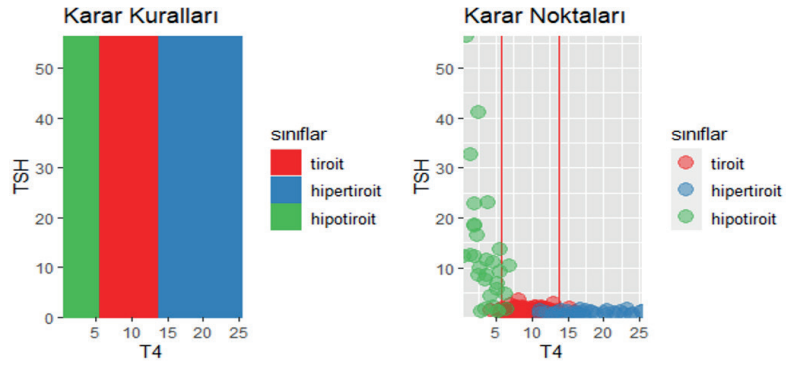
3.1. Tekil Ağaçların Yarattığı Karar Yapıları

Makine öğrenmesi modellerinde şeffaflık, modelin tahmini hangi mantıksal silsile ile gerçekleştirdiğinin kullanıcı tarafından izlenebilmesi anlamına gelir. Bu çalışmada uygulanan tekil karar ağaçları (CART, C5.0 ve CTREE), "içsel olarak açıklanabilir" (intrinsically interpretable) modellerin en güçlü örneklerini teşkil etmektedir. Bu modellerin UCI tiroit veri seti üzerindeki en büyük avantajı, karmaşık biyokimyasal etkileşimleri, tıp doktorlarının aşina olduğu "karar akış şemalarına" (flowcharts) dönüştürebilme yeteneğidir.

UCI verisi üzerinde yapılan analizlerde, özellikle CTREE ve C5.0 modellerinin hiyerarşik yapısı, tiroit fonksiyon bozukluklarının fizyolojik doğasıyla uyum sergilemeye daha uygun olarak nitelendirilebilir. Örneğin, Şekil 1'deki C5.0 çıktısında görüldüğü üzere, model en yüksek istatistiksel anlamlılığa sahip olan T4 ve TSH değişkenlerini kök ve ana düğümler olarak seçmiştir.



Şekil 1. UCI verisi için C5.0 ağacı.



Şekil 2. UCI verisi CTREE koşullu ağaca göre karar analizi.

Topluluk modellerinin (RF, XGBOOST) aksine, bu kurallar herhangi bir matematiksel transformasyon gerektirmeden klinik dile tercüme edilebilir. Bu şeffaflık, hatalı bir tanı durumunda hekimin modelin hangi parametre veya eşik değer nedeniyle yanıldığını saptamasına ve müdahale etmesine olanak tanır.

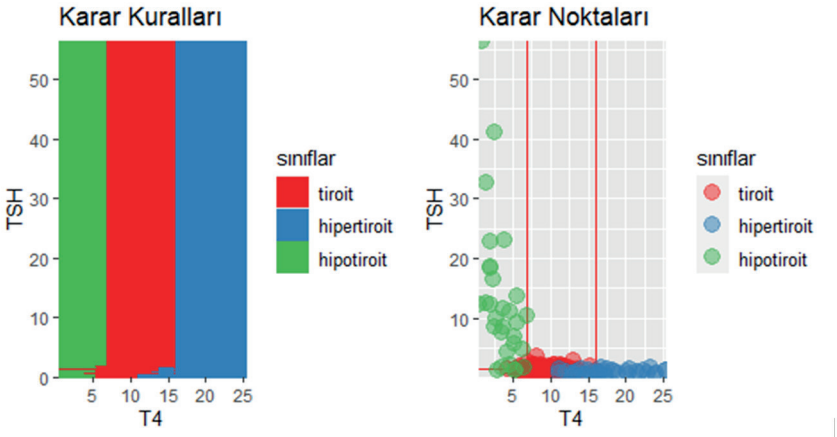
Karar sınırlarının (decision boundaries) görselleştirilmesi, tekil ağaçların şeffaflığını geometrik olarak da kanıtlamaktadır. Şekil 2'de CTREE ağacının görselleştirilmesine örnek teşkil etmektedir. Bu tekil ağacın veriyi geniş ve doğrusal bloklar halinde (sadece T4 hormonuna dayalı) sınıflandırdığı görülmektedir. Makine öğrenmesi modellerinin ürettiği bu geometrik sadelik, klinik rehberlerin (guidelines) sunduğu tıbbi gerçeklikle her zaman tam bir uyum sergilememektedir; çünkü rehberler geniş popülasyon verilerine ve evrensel tıbbi referanslara dayanırken, tekil karar ağaçları sadece eldeki kısıtlı veri setindeki (UCI verisi gibi) matematiksel bölünme noktalarına odaklanmaktadır. CTREE gibi modellerin "fazla sadeleşerek" verideki kritik klinik nüansları kaçırması ve düşük doğruluk oranları vermesi, algoritmik sadeliğin her zaman klinik hassasiyeti temsil etmediğini kanıtlamaktadır (Tablo 1).

Tablo 1. Çeşitli ağaç temelli algoritmalar için test performansı

Model	Parametre/Sınıf	Tiroit	Hipertiroit	Hipotiroit
CTREE	Hassasiyet	0.9667	0.7143	0.8333
	Özgüllük	0.7692	0.9722	1.0000
	F1	0.9355	0.7692	0.9091
	Dengelenmiş Doğruluk	0.8679	0.8433	0.9167
Rastgele Orman	Hassasiyet	1.000	1.0000	0.5000
	Özgüllük	0.7692	1.0000	1.0000
	F1	0.9524	1.0000	0.6666
	Dengelenmiş Doğruluk	0.8846	1.0000	0.7500
XGBOOST	Hassasiyet	1.0000	0.7143	0.6666
	Özgüllük	0.6923	1.0000	1.0000
	F1	0.9375	0.8333	0.8000
	Dengelenmiş Doğruluk	0.8462	0.8571	0.8333
C5.0	Hassasiyet	1.0000	0.7143	0.6666
	Özgüllük	0.6923	1.0000	1.0000
	F1	0.9375	0.8333	0.8000
	Dengelenmiş Doğruluk	0.8462	0.8571	0.8333
CART	Hassasiyet	0.9333	1.0000	0.5000
	Özgüllük	0.7692	0.9444	1.0000
	F1	0.9180	0.8750	0.6666
	Dengelenmiş Doğruluk	0.8513	0.9722	0.7500

Tablo 2. Çeşitli ağaç temelli algoritmalar için genel doğruluk ve kappa açısından test performansı

Model/Parametre	Doğruluk	%95 Güven Aralığı	Kappa
CTREE	0,9070	0,7786 - 0,9741	0,7895
Rastgele Orman	0,9302	0,8094 - 0,9854	0,8371
XGBOOST	0,9070	0,7786 - 0,9741	0,7766
C5.0	0,9070	0,7786 - 0,9741	0,7766
CART	0,8837	0,7492 - 0,9611	0,7434



Şekil 3. UCI verisi C5.0 koşullu ağaca göre karar analizi.

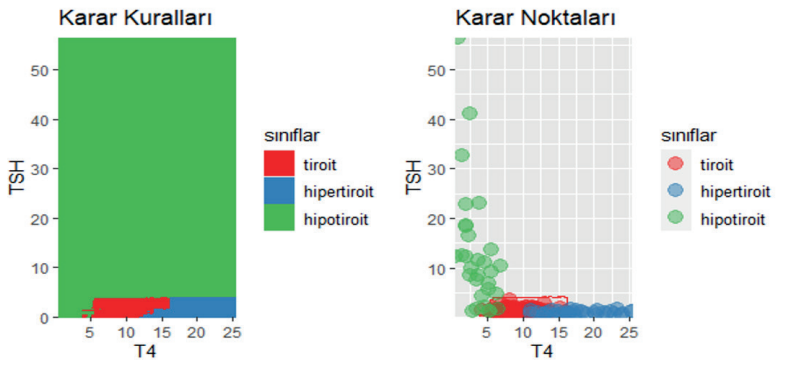
Şekil 3'de C5.0 algoritmasının UCI veri seti üzerindeki karar mekanizması, hem yüksek tahmin gücü hem de tıbbi kılavuzlarla olan görsel uyumu bakımından en ideal "açıklanabilir" model yapısını sergileme iddiasındadır. Karar kuralları grafiğinde görülen geniş ve doğrusal renk blokları, algoritmanın T4 ve TSH değişkenleri üzerinde belirlediği eşik değerlerinin, klinik tanı protokollerindeki hormon referans aralıklarıyla benzer bir mantıkla oluşturulabileceğini gösterir. Karar noktaları grafiği ile birlikte değerlendirildiğinde, modelin XGBOOST veya Rastgele Orman gibi karmaşık yöntemlerin aksine parçalı "adacıklar" oluşturmadığı, verideki gürültüyü ezberlemek yerine (overfitting'den kaçınarak) hastalıkların temel biyokimyasal örüntülerini rasyonalize etmeye daha uygun olabileceği söylenebilir. Sonuç olarak C5.0, hekimlerin bir bakışta takip edebileceği kadar sade bir karar geometrisi sunarak, yüksek doğruluk oranını klinik olarak denetlenebilir bir şeffaflıkla birleştirmeye aday bir modeldir.

3.2. Topluluk Modellerinin Geometrik Karmaşıklığı

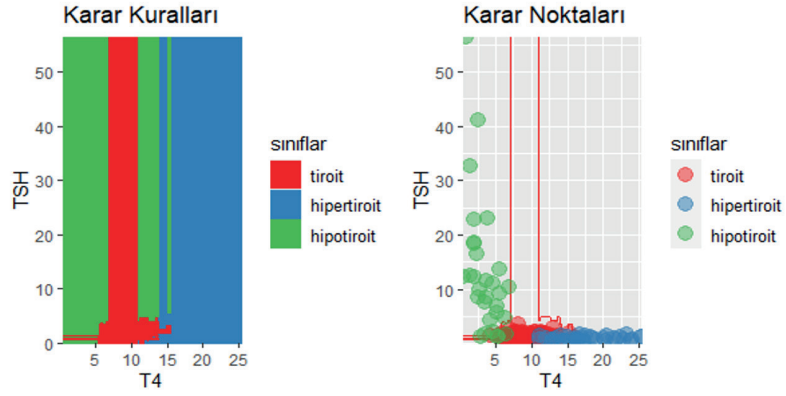
Tablo 1-2, modellerin istatistiksel başarısı ile klinik teşhis kabiliyetleri arasındaki kritik farkları ortaya koymaktadır. Tablo verilerine göre, Rastgele

Orman ve CART modelleri hipertiroit sınıfında %100 hassasiyet (1.0000) ile kusursuz bir performans sergilerken, teşhisi daha zor olan hipotiroit sınıfında bu oran %50'ye (0.5000) düşerek vakaların yarısının gözden kaçırılmasına neden olmuştur. Buna karşın CTREE modeli, %90,70 genel doğruluk oranı sergilemesine rağmen hipotiroit sınıfında ulaştığı %83,33 hassasiyet ve %91,67 dengelenmiş doğruluk değeriyle, sınıflar arası performansı en iyi normalize eden ve klinik açıdan en güvenilir sonuçları veren yöntem olarak öne çıkmaktadır. XGBOOST ve C5.0 modelleri ise tüm metriklerde birbirine paralel ve orta düzeyde bir başarı sergileyerek daha stabil bir profil çizmiştir.

Modellerin test aşamasındaki bu sınıf bazlı ayrışması, aşırı öğrenme (overfitting) riskinin de somut bir göstergesidir. Rastgele Orman ve XGBOOST gibi topluluk modelleri genel doğrulukta %93,02 ile zirveye çıksa da, azınlık sınıflardaki (hipotiroit) düşük hassasiyetleri, bu modellerin eğitim verisindeki baskın grupları ezberlediğini ve verideki gürültüye odaklandığını kanıtlamaktadır. Buna karşılık CTREE'nin sergilediği dengeli dağılım, modelin veriyi ezberlemek yerine gruplar arasındaki temel istatistiksel örüntüleri daha iyi genellediğini ve aşırı öğrenmeden kaçınarak klinik gerçekliğe daha yakın sonuçlar ürettiğini doğrulamaktadır. Bu durum, tiroit gibi hassas tanılarda sadece yüksek genel doğruluk oranının değil, sınıflar arası hassasiyet dengesinin de bir başarı kriteri olması gerektiğini ortaya koymaktadır.



Şekil 4. UCI verisi rastgele orman modeline göre karar analizi.



Şekil 5. UCI verisi XGBOOST modeline göre karar analizi.

Şekil 4, Rastgele Orman algoritmasının UCI veri seti üzerindeki karar mekanizmasını yansıtmakta olup; modelin istatistiksel başarısı ile klinik yorumlanabilirliği arasındaki "açıklanamazlık" sorununu somutlaştırmaktadır. Karar kuralları grafiğinde hipertiroit (mavi) ve tiroit (kırmızı) sınıflarının kesiştiği alt bölgedeki sınırların doğrusal olmayışı ve parçalı yapısı, modelin verideki mikro örüntüleri yakalamak adına tıbbi rehberlerde yer alan basit eşik değerlerinden uzaklaşan girift bir geometri oluşturduğunu kanıtlamaktadır. Karar noktaları grafiği ise hipotiroit (yeşil) vakalarının düşük T4 ve geniş bir TSH aralığında dağıldığını, modelin ise gözlem yapılmayan alanları dahi bu sınıfa dahil ederek aşırı öğrenme (overfitting) riski taşıyan keskin bir genelleme yaptığını göstermektedir. Sonuç olarak, bu karmaşık yapı her ne kadar test verisinde yüksek bir doğruluk oranına ulaşsa da, sınıf geçişlerindeki belirsizlik ve görsel kaos, hekimlerin algoritma kararlarını rasyonalize etmesini zorlaştırarak klinik güveni zedeleyebilecek bir "kara kutu" karakteri sergilemektedir.

Şekil 5, XGBOOST algoritmasının UCI veri seti üzerindeki karar mekanizması, modelin yüksek tahmin gücü ile düşük görsel açıklanabilirliği arasındaki çelişkiyi açıkça ortaya koymaktadır. Karar kuralları grafiğinde özellikle tiroit (kırmızı) ve hipertiroit (mavi) sınıfları arasında gözlemlenen dikey şeritli ve çok parçalı yapı, algoritmanın ardışık hataları düzeltmek amacıyla oluşturduğu mikro karar bölgelerini yansıtmaktadır. Karar noktaları grafiği ile birlikte değerlendirildiğinde, modelin düşük T4 ve geniş bir TSH aralığındaki vakaları keskin bir şekilde hipotiroit olarak tanımladığı, ancak veri bulunmayan boşluklarda dahi bu sert geçişleri sürdürerek aşırı öğrenme (overfitting) riski taşıyan bir geometri çizdiği görülmektedir. Sonuç olarak XGBOOST, test verisinde başarılı bir doğruluk oranına ulaşsa da, ürettiği bu parçalı ve karmaşık karar sınırı yapısı nedeniyle hekimlerin teşhis adımlarını basit bir akış

şemasıyla takip etmesini zorlaştıran bir "kara kutu" karakteri sergilemektedir.

4. SONUÇ VE ÖNERİLER

UCI veri seti üzerinde gerçekleştirilen bu çalışma, makine öğrenmesi modellerinin tıbbi tanı süreçlerindeki başarısının sadece istatistiksel doğruluk oranlarıyla değil, aynı zamanda klinik açıklanabilirlik ve karar sınırlarının geometrik sadeliğiyle değerlendirilmesi gerektiğini ortaya koymuştur. Analiz sonuçları; Rastgele Orman algoritmasının doğruluk oranı ile genel performansta zirvede yer aldığını, ancak hipotiroit gibi teşhisi kritik sınıflarda %50 hassasiyet değerinde kalarak aşırı öğrenme (overfitting) eğilimi gösterdiğini doğrulamaktadır. Buna karşın CTREE modelinin, genel doğruluğu %90,70 olmasına rağmen hipotiroit sınıfında ulaştığı %83,33 hassasiyet ve %91,67 dengelenmiş doğruluk değeriyle, sınıflar arası performansı en iyi ayıran ve klinik açıdan en güvenilir sonuçları sunan yöntem olmaya aday olduğu söylenebilir.

Görsel analizler, modellerin karar verme mekanizmaları ile tıbbi gerçeklik arasındaki uyumu net bir şekilde yansıtmayabilir. Rastgele Orman ve XGBOOST gibi topluluk modellerinin ürettiği parçalı, dikey şeritli ve adacıklar içeren karmaşık karar sınırları, bu algoritmaların verideki gürültüye odaklanarak tıbbi literatürde karşılığı olmayan mikro karar bölgeleri oluşturduğunu kanıtlamaktadır. Bu geometrik kaos, yüksek tahmin gücü sağlasa da hekimlerin algoritma kararlarını rasyonalize etmesini zorlaştırmaktadır. Öte yandan, C5.0 gibi modellerin sunduğu daha sade ve doğrusal bloklardan oluşan karar geometrisi, hem aşırı öğrenme riskini minimize etmekte hem de klinik rehberlerdeki hormon eşik değerleriyle daha tutarlı bir görsel çerçeve sunmaktadır.

Sonuç olarak, tıbbi karar destek sistemlerinin tasarımında sadece en yüksek genel doğruluk skoruna odaklanmak yerine, teşhis edilmesi zor vakalarda yüksek hassasiyet sunan ve denetlenebilir bir mantık akışı sergileyen modellerin tercih edilmesi kritik önemdedir. Gelecekteki uygulamalarda, yapay zeka sistemlerinin sunduğu "karar kuralları" ve "geometrik sınırların" hekimlerin denetimine açılması, teknoloji ile tıp profesyonelleri arasındaki güven bağını güçlendirecektir. Bu bağlamda, performans ve açıklanabilirlik arasında optimal bir denge kuran C5.0 ve CTREE gibi algoritmaların, klinik standartlara ve etik sorumluluklara daha uygun bir zemin hazırladığı değerlendirilmektedir.

Veri setinde sınıfların 150 Tiroit, 35 Hipotiroit ve 30 Hipertiroit şeklinde dağılması, makine öğrenmesi literatüründe "dengesiz veri seti" (imbalanced dataset) problemi olarak tanımlanmakta ve model performansları üzerinde doğrudan belirleyici bir rol oynamaktadır. Bu dengesizlik, modellerin baskın olan "Tiroit" (normal) sınıfına karşı bir yanlılık (bias) geliştirmesine yol açarak genel doğruluk oranlarını yapay olarak yükseltmekte; ancak azınlıkta

kalan hastalık sınıflarında, özellikle Rastgele Orman ve CART gibi modellerin hipotiroit tanısında 0.5000 gibi düşük hassasiyet (sensitivity) değerlerinde kalmasına neden olmaktadır. Sayıca az olan sınıflardan genel kurallar çıkarmak yerine bu örneklerin gürültüsünü ezberlemeye çalışan XGBOOST gibi karmaşık modellerde aşırı öğrenme (overfitting) riski artarken, CTREE ve C5.0 modellerinin bu veri kısıtına rağmen daha dengeli bir performans sergilemesi, bu algoritmaların istatistiksel olarak daha sağlam ve genellenebilir karar yapıları üretebildiğini kanıtlamaktadır.

Veri setindeki dengesizliği gidermek amacıyla kullanılan sentetik veri artırımı (over-sampling) gibi yöntemlerin tercih edilmemesi, bu tür yaklaşımların klinik uygulama açısından yaratabileceği ciddi risklerden kaynaklanmaktadır. Sentetik veri üretimi, üretim miktarına bağlı olarak, gerçek biyokimyasal süreçlerin bir sonucu olmayan "yapay" hasta kayıtları oluşturarak klinik gerçeklikten kopulmasına ve modelin gerçek dünya verileri üzerindeki güvenilirliğinin zedelenmesine yol açabilmektedir. Özellikle tıbbi teşhis süreçlerinde, azınlık sınıfı örneklerinin kopyalanması modelin bu sınırlı vakaları ezberlemesine (overfitting) neden olarak, hekimlerde modelin genelleme yeteneği hakkında yanıltıcı bir güven duygusu oluşturabilmektedir. Ayrıca, çoğunluk sınıfından veri silinmesi (under-sampling) ise sağlıklı bireylere ait kritik klinik örüntülerin ve değişkenler arası doğal varyasyonun kaybedilmesi anlamına gelebilir. Bu nedenlerle, veriye yapay müdahalelerle "ideal" bir dağılım yaratmak yerine, mevcut dengesizliğe rağmen istatistiksel olarak sağlam (robust) kararlar üretebilen CTREE gibi modellerin kullanılması ve performansın "Dengelenmiş Doğruluk" gibi şeffaf metriklerle raporlanması uygun bir yaklaşım olarak değerlendirilmiştir.

Referanslar

Akgül, G., Çelik, A. A., Aydın, Z. E., ve Öztürk, Z. K. (2020). Hipotiroidi Hastalığı Teşhisinde Sınıflandırma Algoritmalarının Kullanımı. *Bilişim Teknolojileri Dergisi*, *13*(3), 255-268.

Banu G. R. (2016). A Role of decision Tree classification data Mining Technique in Diagnosing Thyroid disease. *International Journal of Computer Sciences and Engineering*, *4*(11), 64-70.

Begum A. ve Parkavi A. (2019). Prediction of thyroid disease using data mining techniques. *In 2019 5th International Conference on Advanced Computing and Communication Systems* (ICACCS) (pp. 342-345).

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Kousarrizi M. N., Seiti F., ve Teshnehlab M. (2012). An experimental comparative study on thyroid disease diagnosis based on feature subset selection and classification. *International Journal of Electrical ve Computer Sciences IJECS-IJENS*, *12*(01), 13-20.

Margret J., Lakshmipathi B., ve Kumar S. A. (2012). Diagnosis of thyroid disorders using decision tree splitting rules. *International Journal of Computer Applications*, *44*(8), 43-46.

Quinlan, R. (1986). Thyroid Disease [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5D010.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

Chen, T. (2016). XGBoost: A Scalable Tree Boosting System. *Cornell University*.

Bölüm 5

KARAYOLLARI MOTORLU ARAÇLAR ZORUNLU MALİ MESULİYET SİGORTALARINDA AKTÜERYAL PRİM HESABI

Buse Badatlı[1]

Pelin Kasap[2]

1 (Yüksek Lisans Mezunu), Ondokuz Mayıs Üniversitesi, Fen Fakültesi, İstatistik Bölümü, Samsun, Türkiye,e-mail: badatlibuse@gmail.com, ORCID:0000-0001-6805-5874

2 (Prof. Dr.) Ondokuz Mayıs Üniversitesi, Fen Fakültesi, İstatistik Bölümü, Samsun, Türkiye, e-mail: pelin.kasap@omu.edu.tr , ORCID:0000-0002-1106-710X

Bu çalışma, yüksek lisans tezinden üretilmiştir. Tezin künye bilgileri: Badatlı, B. (2024). Trafik ve Sağlık Sigortası Ürünlerinde Genelleştirilmiş Lineer Modeller ile Aktüeryal Prim Hesabı, Yüksek lisans tezi, Danışman: Prof. Dr. Pelin Kasap, Ondokuz Mayıs Üniversitesi, Lisansüstü Eğitim Enstitüsü, Samsun.

1. Giriş

Sigorta şirketleri yaşanabilecek riskleri teminat altına alan finansal kurumlardır ve teminat altına aldığı riski doğru fiyattan alıp satabilmek en öncelikli politikalarındandır. Sigorta şirketleri, benzer risk profiline sahip sigortalıları belirli bir prim karşılığında portföylerinde toplayarak, bu sigortalıların risklerini üstlenmekte ve onlara güvence sağlamaktadır. Ancak, sigorta şirketlerinin öncelikli amacı, güvence sunmakla birlikte, her ticari kurum gibi kâr elde ederek sürdürülebilirliklerini sağlamaktır (Kaas vd., 2001). Ticaret sektöründe, bir ürünün satış fiyatı, maliyetine eklenen kâr ile belirlenirken, sigorta alanında poliçe satışı sırasında hasarın henüz gerçekleşmemesi maliyetin belirsiz hale gelmesine yol açmakta ve bu durum da prim hesaplamasını zorlaştırmaktadır. Bu nedenle, sigorta şirketlerinin olası hasarları karşılayabilmesi ve aynı zamanda kâr elde edebilmesi için primlerini aktüeryal yöntemlerle mümkün olan en doğru biçimde hesaplamaları kritik öneme sahiptir. Bu hesaplamaların gerçekleştirilmesi için, sigorta şirketinin karşılaşabileceği iki ana risk unsuru olan hasar tutarı ve hasar sıklığı doğru bir şekilde öngörülmelidir. İlgili öngörü süreçlerinde ise istatistiksel dağılımlardan yararlanılmaktadır (Szegö, 2004).

Şahin vd. (2016) Türkiye'deki çeşitli coğrafi tehlike bölgeleri çerçevesinde buğday bitkisel ürününe yönelik tarım sigortası kapsamındaki hasar tutarlarını inceleyerek, her bir tehlike bölgesine ilişkin hasar tutarı dağılımını belirlemiş ve aktüeryal prim hesaplama yöntemlerini kullanarak prim hesaplamaları gerçekleştirmiştir. Gültekin ve Erdemir (2010) yangın sigortası primini tahmin edebilmek amacıyla bir demir-çelik şirketinden temin ettikleri verilerle yangın hasarlarına ilişkin hasar tutarı ve hasar sıklığı dağılımlarını elde ederek aktüeryal prim hesaplamalarını gerçekleştirmişlerdir. Selimović (2010) gerçekleştirdiği çalışmada aktüerlerin prim hesaplamalarının yanı sıra, şirketlerin ticari kâr elde etmek amacıyla gelirlerinin bir kısmını teknik karşılıklar olarak ayırma çabası içerisinde olduklarını ifade etmiştir. Teknik karşılıkların doğru yöntemlerle belirlenmesi, şirketin ödeme gücünü etkileyip bunun sonucunda doğru teknik karşılık ayrılmasının şirketin güvenilirliği üzerindeki etkisini vurgulamakta ve bu bağlamda uygun aktüeryal yöntemler kullanılarak teknik karşılıkların en doğru biçimde tespit edilmesi gerekliliğine dikkat çekmektedir. Frees ve Huang (2023) aktüerlerin sigorta fiyatlandırmasında sigortalıları sınıflandırma gerekliliğini tartışmış ve bu ayrımın uygunluğunu değerlendirmek amacıyla sosyal ve ekonomik ilkeleri incelemiştir.

Bu çalışmada, Türkiye'de zorunlu bir sigorta türü olarak öne çıkan ve sektörde en yüksek poliçe ile prim üretimine sahip olan trafik sigortası olarak da bilinen karayolları zorunlu mali mesuliyet sigortasına ait veriler incelenmiştir. Aktif olarak faaliyet gösteren bir sigorta şirketinden 2016-2019

yılları arasında kamyon kullanım tarzına yönelik poliçeleşen 10244 adet gerçek sigortalı verisi kullanılmıştır. Çalışmanın ilk aşamasında, prim hesabına yönelik dağılımların belirlenmesi gerçekleştirilmiş ve dağılımların parametreleri, parametre tahminlerinde sıkça başvurulan yöntemlerden biri olan En Çok Olabilirlik (EÇO) yöntemi kullanılarak belirlenmiştir (Bain and Engelhardt, 1992). Dağılımların ve dağılım parametrelerinin belirlenmesinin ardından, Genelleştirilmiş Lineer Modeller (GLM) yardımıyla değişkenlere ait katsayılar belirlenmiş ve böylece prim hesabına yönelik bir tarife denklemi oluşturulmuştur.

Çalışmanın geri kalan bölümleri aşağıdaki şekilde yapılandırılmıştır: Tarife modelinin oluşturulmasında hangi yöntemin kullanılacağının belirlenmesi önemli bir adım olduğundan, bireysel ve kolektif risk modelleri 2. bölümde ele alınmıştır. 3. bölümde, hasar tutarı ve hasar sıklığına ait dağılımlar hakkında bilgi verilmiştir. 4. bölümde, elde edilen dağılımlar kullanılarak prim belirlemek amacıyla GLM hakkında bilgi verilmiş, ardından 5. bölümde bulgular ve tartışmaya yer verilmiştir. Çalışmadan elde edilen sonuçlar 6. bölümde sunulmuştur.

2. Bireysel ve Kolektif Risk Modelleri

Sigortacılıkta şirketlerin karşılaşabileceği toplam hasar maliyeti, portföylerinde bulunan sigortalılarının neden olduğu hasarların toplamı şeklinde tanımlanabilir. Bu hasar maliyetinin modellenmesinde iki farklı yaklaşım kullanılmaktadır. Bu yaklaşımlar, bireysel risk modeli ve kolektif risk modelidir (Tse, 2009).

Bireysel Risk Modeli

Sigorta şirketinin portföyüne ait sigortalıların adedi n ve $i = 1, \cdots, n$ için i. hasar tutarı ise X_i ile temsil edilsin. Bu bağlamda, her bir hasar tutarının (X_i'nin) birbirinden bağımsız ve aynı dağılıma sahip olduğu varsayımıyla hareket edilmektedir. Bu çerçevede, portföyün toplam hasarı (1) eşitliği ile hesaplanmaktadır.

Burada toplam hasar S'nin, aynı dağılıma sahip n tane bağımsız değişkenin toplamı olduğu görülmektedir. Sigorta portföyünün tamamı için hesaplamalarda bu risk modeli yeterli olmayabilir; ancak bireysel düzeyde derecelendirme yapılabilmesi açısından önemli bir role sahiptir (Denuit vd., 2005).

$$S = X_1 + \cdots + X_n \tag{1}$$

Kolektif Risk Modeli

Kolektif risk modelinde, toplam hasarın hesaplanabilmesi için i'inci hasara ait tutar X_i ve portföydeki poliçelerin belirli bir periyottaki hasar sıklığı N rastgele değişkenlerinin dağılımlarının bilinmesi gerekmektedir. Toplam

hasar kolektif risk modeli yardımıyla (2) eşitliğindeki gibi hesaplanır.

$$S = X_1 + \cdots + X_N \quad (2)$$

Kolektif risk modelinde, X_i rastgele değişkenleri aynı dağılıma sahip, N ve X_i rastgele değişkenlerinin de bağımsız olduğu varsayımları bulunmaktadır. Bu durum, toplam hasar etkilerini daha açık bir şekilde analiz etmeye olanak tanımaktadır (Şahin vd., 2016). Bu yöntemde, portföye ait toplam poliçe sayısı değil, yalnızca hasar oluşturan poliçeler hesaba katılmaktadır (Tse, 2009).

3. Hasar Tutarı ve Hasar Sıklığı Dağılımları

Sigortacılıkta şirketlerin sahip olduğu ödeme yükümlülüğü, sigortalı ile sigorta şirketi arasındaki sözleşmeye dayalı ilişki çerçevesinde tanımlanmaktadır ve sigortalıya ait koruma altına alınmış riskin meydana gelmesi halinde ortaya çıkmaktadır. Bahsi geçen risk, sigortalıların hasar tutarları ve hasar sıklıklarının birbirinden bağımsız olduğuna dayanmaktadır. Hasar sıklığını öngörmek önemli olmakla birlikte, yalnızca bu tahmin yeterli değildir; çünkü tek başına maliyetin büyüklüğünü ifade etmemektedir. Dolayısıyla, hasar tutarının da doğru bir şekilde tahmin edilmesi gerekmektedir. Sigorta şirketleri, tutar ve sıklığı doğru öngörebildiklerinde toplam maliyet hakkında bir değerlendirme yapma imkânına sahip olurlar. Aktüerler, hasar tutarını modellemek için sürekli dağılımları, hasar sıklığını modellemek için ise kesikli dağılımları kullanmaktadırlar. Bu modellemelerde, sıklık ve tutarın negatif olamayacağı dikkate alınarak, pozitif dağılımlar tercih edilmektedir (Bahnemann, 2015). Ayrıca, aktüerler istekleri doğrultusunda bu ayrı iki dağılımı tek bir dağılım altında birleştirerek yönetme seçeneğine de sahiptirler (Tse, 2009).

Hasar sıklığına ait dağılımlar, sigorta ve risk yönetimi alanında önemli bir yer tutar. Negatif olmayan tamsayılar üzerinde tanımlanan bu dağılımlar, genellikle hasar olaylarının sayısını modellemek için kullanılır. Poisson, Binom, Geometrik ve Negatif Binom dağılımları bu alanda en yaygın kullanılan dağılımlar arasında yer alır. Poisson dağılımı, belirli bir zaman aralığında gerçekleşen olayların sayısını modellemek için idealdir. λ parametresi, olayların ortalama sayısını temsil eder ve bu değer, zaman dönemine göre ayarlanabilir. Bu esneklik, hasar sıklığı verilerine yüksek uyum sağlamasına olanak tanır. Binom dağılımı, belirli sayıda denemede başarıların sayısını modellemek için kullanılır. Varyansın ortalamadan küçük olduğu durumlarda uygun bir tercih olur. Sigorta portföylerinde, genellikle sınırlı sayıda hasar olayı gözlemlendiğinde tercih edilir. Negatif Binom dağılımı, Poisson dağılımına göre daha fazla esneklik sunar. İki parametreye sahip olması, daha ağır kuyruklar ve daha değişken veri setleri ile başa çıkabilmesini sağlar. Özellikle hasar olaylarının sıklığının değişkenlik gösterdiği durumlarda tercih edilir. Bu dağılımların seçimi, hasar verilerinin

özelliklerine ve portföyün yapısına bağlı olarak değişir. Doğru modelin seçilmesi, risk yönetimi ve prim hesaplamalarında önemli bir rol oynar (Klugman vd., 2012).

Hasar tutarları, sigortacılıkta şirketlerin yükümlülüklerini belirlemede kritik bir rol oynamakta olup, bu tutarların modellenmesi için uygun dağılımların seçimi büyük önem taşımaktadır. Hasar tutarlarını etkileyen kuyruk yapıları, küçük veya büyük çapta hasarların meydana gelebileceğini gösterdiğinden, uzun kuyruklu dağılımlar tercih edilmektedir. Gamma dağılımı, iki parametreli yapısı sayesinde sağladığı esneklik ile hasar tutarlarının modellenmesinde yaygın olarak kullanılmaktadır. Pareto dağılımı, uzun kuyruk özelliği sayesinde büyük hasarların olasılığını artırarak aktüerler tarafından sıkça tercih edilirken, hesaplamalarda kolaylık sağlamak amacıyla tek parametreli hale getirilebilmektedir. Log-normal dağılım ise pozitif çarpıklığı ile, hasar tutarlarının doğal olarak pozitif değerler almasını yansıtır; bu durum, küçük hasarların daha sık ortaya çıkma olasılığını göstermektedir. Diğer dağılımlar arasında üstel dağılım, kısa kuyruklu özellikleri nedeniyle büyük hasarların modellenmesinde sınırlı kalırken, Weibull dağılımı farklı kuyruk yapılarında hasar tutarlarının analizine olanak tanımaktadır. Sonuç olarak, hasar tutarlarının doğru bir biçimde modellenmesi, sigorta şirketlerinin risk yönetimi ve prim belirleme süreçlerinde hayati bir öneme sahiptir (Bahnemann, 2015).

4. Genelleştirilmiş Lineer Model

Ekonomik riskin sigortalıdan sigortacıya devredilmesi, sigorta şirketlerinin hasar yönetimi ve prim hesaplamaları üzerinde önemli bir etkiye sahiptir. Bağımsız küçük hasarların bir araya gelmesi, büyük sayılar kanunu gereği toplam kaybın daha öngörülebilir hale gelmesini sağlamaktadır. Bu durum, sigorta primlerinin hesaplanmasında beklenen hasarların temel alınmasını zorunlu kılmaktadır.

Ancak, sigorta şirketlerinin farklı risk gruplarını adil bir şekilde fiyatlandırmaması olumsuz seçilim sorununu doğurabilir. Düşük riskli grupların düşük primlerle sigorta yaptırması, yüksek riskli grupların ise daha yüksek primlerle karşılaşmasına yol açabilir. Bu da yüksek primli grubu alternatif sigorta şirketlerine yönlendirebilir.

Rekabetçi piyasalarda adil prim talep etmek, şirketlerin sürdürülebilirliği açısından kritik bir öneme sahiptir. Doğru risk değerlendirmesi ve beklenen kayıpların sağlıklı bir şekilde tahmin edilmesi, riskin etkili bir şekilde paylaştırılmasını ve istatistiksel modellere dayanan prim hesaplamalarının uygulanmasını mümkün kılar. Bu bağlamda, genelleştirilmiş lineer modellerin kullanımı, sigorta şirketlerinin daha sağlam kararlar almasına yardımcı olabilir ve uzun vadede ekonomik kayıpların minimize edilmesine katkıda bulunabilir.

Sigorta şirketlerinin fiyatlandırma stratejilerini oluştururken, sektördeki rekabetçi konumlarını korumak ve portföydeki sigortalıları ayrıştırmak adına ayrıştırıcı primler geliştirmeleri gerekmektedir. Bu süreçte, sigortalıları etkili bir şekilde farklılaştırmak için gerekli teknik donanıma sahip olmak büyük önem taşır. Fiyatlandırma sürecinin gerçekleştirileceği her bir branş (sağlık, trafik, konut, kasko vb.) için spesifik faktörler (yaş, cinsiyet, il vb.) her zaman mevcut olmayabilir; bu nedenle, uygun faktörlerin tespit edilmesi ve bu faktörlerin modellenen riskle ilgili olup olmadığının belirlenmesi kritik bir ihtiyaç haline gelmektedir. Genelleştirilmiş lineer modeller (GLM), istatistiksel analizde geniş bir uygulama yelpazesine sahip olup, klasik doğrusal regresyonun ötesine geçerek çeşitli dağılımlara sahip yanıt değişkenleri ile ilişkilerin incelenmesine olanak tanır (Nelder ve Wedderburn, 1972). Bu modeller, doğrusal ilişkileri modellemenin yanı sıra, yanıtın dağılımını belirlemek için link fonksiyonları kullanarak esneklik sunar. Bu nedenlerle GLM, en yaygın olarak kullanılan yöntemlerden biridir. Daha önce, tarifede uygun faktörleri belirlemek için minimum varyans yöntemi gibi eski teknikler uygulanmaktaydı (Mildenhall, 1999). Ayrıca, geçmişte tek yönlü ve iki yönlü analiz teknikleri de kullanılmaktaydı. Tek yönlü analiz, yaş faktörünün modele uygunluğunu değerlendirirken; iki yönlü analiz, yaş ve cinsiyet gibi iki faktör arasındaki ilişkiyi incelemektedir. Çok yönlü analizlerde ise, birden fazla faktörün uygunluğu sistematik bir şekilde değerlendirilmektedir. Ancak, faktör sayısının artmasıyla birlikte çok yönlü analizin uygulanması güçleşebilmektedir (Parodi, 2023). Tüm bu yöntemler, tarifeye katkı sağlayacak faktörlerin regresyon denklemindeki bağımlı değişkene etkisini tespit etmeyi amaçlamaktadır. GLM analizleri için sektörde pek çok yazılım paketi mevcuttur ve bu araçlar aracılığıyla tarife analizleri kolaylıkla gerçekleştirilebilmektedir (Ohlsson ve Johansson, 2010).

4.1. Çarpımsal Modeller

Tarife analizi, genellikle sigortacının mevcut veri setlerine dayanarak gerçekleştirilmektedir. Her bir tarife için yeterli hasar verisi mevcut olduğunda, beklenen maliyet, yalnızca gözlemlenen saf prim üzerinden tahmin edilebilir. Ancak, belirli dönemlerde tek bir hasar dahi gerçekleşmediğinde, bu yöntemle prim belirlemek mümkün olmamaktadır. Bu nedenle, her sigortalı için beklenen saf primin doğru bir şekilde tahmin edilmesini sağlayan yaklaşımlara ihtiyaç vardır.

Primlerin zaman içinde istikrarlı kalması ve büyük rastgele dalgalanmalara karşı dayanıklı olması önemlidir; bu durum, beklenen saf primin bir dizi derecelendirme faktörüne bağlılığını gösteren modeller aracılığıyla sağlanır. Çarpımsallık varsayımı, derecelendirme faktörleri arasında herhangi bir etkileşim olmadığını ifade eder (Ohlsson ve Johansson, 2010).

5. Bulgular ve Tartışma

Sektörde etkin bir şekilde faaliyet gösteren bir sigorta şirketinden temin edilen 2016-2019 yılları arasındaki 10244 kamyon ürünü verisi, sigortalılara ait yaş, cinsiyet, bölge, hasar adetleri ve hasar tutarları bilgilerini içermektedir. Bu veri seti içerisinde, 9406 adet erkek ve 838 adet kadın poliçesi bulunmaktadır. Coğrafi bölge ve yaş değişkenlerine ait hasar sıklıkları ise sırasıyla Tablo 1 ve Tablo 2'de sunulmuştur.

Tablo 1. Kamyon veri seti için coğrafi bölgelere göre hasar sıklıkları

Bölge:	Akdeniz	Doğu Anadolu	Ege	Güneydoğu Anadolu	İç Anadolu	Karadeniz	Marmara
n :	1243	224	1847	294	2375	1373	2888

Tablo 1 incelendiğinde, kamyon poliçe adetlerinin bölgelerin nüfus yoğunluğu ile paralel bir dağılım sergilediği gözlemlenmektedir. Ancak, Doğu Anadolu ve Güneydoğu Anadolu bölgelerinde, şirketin genel ortalamasının altında poliçe adetleri kaydedilmiştir. Bu durum, bölgeler arası poliçe dağılımının yalnızca nüfus ile değil, aynı zamanda şirketin portföy yönetiminde benimsediği tarife stratejisiyle de ilişkili olabileceğini göstermektedir. Örneğin, her bölgedeki sürücü davranışları, otoyol altyapısının durumu ve kamyonların ticari amaçlarla kullanımı gibi faktörler, o bölgelerin ekonomik düzeyi ile birleşerek şirketin tarife stratejisi üzerinde belirleyici bir etki yaratmaktadır.

Tablo 2. Kamyon veri seti için yaş değişkenine göre hasar sıklıkları

Yaş Grubu	: 0-20	21-30	31-40	41-50	51-60	61-70	70+
n	: 46	977	2650	3022	2402	986	161

Tablo 2 incelendiğinde, kamyon poliçe adetlerinin yoğunluğunun 31-60 yaş aralığında yoğunlaştığı görülmektedir. Genel olarak trafik ürünündeki portföyün de poliçe adetleri açısından benzer bir dağılım sergilediği söylenebilir. Ayrıca, ticari amaçlarla kullanılan kamyon portföyüne ait sigortalıların yaşlarına bakıldığında, ağırlığın iş hayatında daha aktif olan yaş grubuna denk geldiği tespit edilmiştir. Bu durum, ticari faaliyetlerin yoğunluğunun belirli yaş gruplarıyla ilişkilendirilebileceğini göstermektedir.

Tablo 3. Kamyon veri seti için hasar sıklığına ilişkin betimleyici istatistikler

n	Min	Ortalama	Max	Medyan	St.Hata	Çarpıklık	Basıklık
10244	0	0,1034	1	0	0,3045	2,6041	4,7824

Tablo 4. Kamyon veri seti için hasar tutarına ilişkin betimleyici istatistikler

n	Min	Ortalama	Max	Medyan	St. Hata	Çarpıklık	Basıklık
10244	0	508,11	59248	0	2558,371	9,0919	117,9848

Tablo 3 ve Tablo 4'te sırasıyla kamyon veri seti için hasar sıklığı ve hasar tutarı için tanımlayıcı istatistikler verilmiştir. Tarife modeline başlamadan önce verinin tanınması aktüerler açısından kritik bir öneme sahiptir.

Tablo 5'te Akaike Bilgi Kriteri (AIC) ve Bayes Bilgi Kriteri (BIC) ve EÇO parametre tahminleri verilmiştir. Hesaplamalarda R-Studio programı kullanılmıştır. Tablo 5'in ilk bölümünde hasar sıklığı dağılımları (Poisson, Negatif Binom, Geometrik), ikinci bölümünde ise hasar tutarı dağılımları (Gamma, Üstel, Pareto, Log-Normal, Weibull) verilmektedir. AIC ve BIC değerleri doğrultusunda, hasar tutarına ait dağılımın Log-Normal, hasar sıklığına ait dağılımının ise Poisson dağılımına uygun olduğu tespit edilmiştir.

Hasar sıklığı ve tutarı için dağılımların ve dağılımlara ait parametrelerin belirlenmesi, model kurma süreci için gerekli ön hazırlığı tamamlamaktadır. Ticari tarifenin elde edilmesi amacıyla, GLM süreci R-Studio programı aracılığıyla gerçekleştirilmiştir. Bu süreçte, trafik ürününe ait kamyon verisi, GLM analizinde tüm kategorileriyle birlikte kapsamlı bir şekilde analiz edilmektedir.

Tablo 5. Kamyon veri setine ait hasar sıklığı ve hasar tutarı için EÇO parametre tahminleri ve AIC ve BIC sonuçları

	$\hat{\mu}$	$\hat{\alpha}$	$\hat{\lambda}$	$\hat{p}$	$-log\ L$	AIC	BIC
Poisson	-	-	0,1034	-	-3464,5	**6931,05**	**6938,29**
N. Binom	0,1034	-	-	-	-3464,5	6933,05	6947,52
Geo.	-	-	-	0,9062	-3517,5	7037,14	7044,37
Gamma	-	0,5792	8477,80	-	-10156,1	20316,33	20326,27
Üstel	-	-	4910,50	-	-10069,0	20140,16	20145,12
Pareto	-	3,9670	-	-	-10032,4	20068,82	20078,75
LogNorm.	7,9182	1,0460	-	-	-9945,14	**19894,28**	**19904,22**
Weibull	-	0,9288	4712,8	-	-10063,2	20130,56	20140,50

Tablo 6. Kamyon veri seti için hasar sıklığına ilişkin GLM sonuçları

	Tahmin	Katsayılar
BAZ	-2,62387	0,07252
0-20	0,00000	1,00000
21-30	0,17063	1,18605
31-40	0,26187	1,29935
41-50	0,00604	1,00606
51-60	0,15257	1,16482
61-70	0,07596	1,07892
70+	-0,19849	0,81997
K	0,27810	1,32062
E	0,00000	1,00000
Akdeniz	0,00000	1,00000
Doğu Anadolu Bölgesi	-0,09579	0,90866
Ege Bölgesi	0,06133	1,06325
Güneydoğu Anadolu Bölgesi	-0,31033	0,73320
Karadeniz Bölgesi	0,04545	1,04650
Marmara Bölgesi	0,55739	1,74610
İç Anadolu Bölgesi	0,00061	1,00061

Tablo 6'da, trafik ürünü kamyon portföyüne ait GLM model sonuçları incelendiğinde, öncelikle sigortalı yaş grubuna ilişkin 0-20 yaş aralığından 40 yaşa kadar olan katsayıların yükseldiği, ancak 40 yaşından sonra 70+ yaş grubuna kadar katsayıların düşüşe geçtiği gözlemlenmektedir. Genel olarak, yaş arttıkça sürüş tecrübesinin de artması beklenirken, hasar sıklığının düşmesi öngörülmektedir. Ancak, bu beklenti ile model sonuçları karşılaştırıldığında, 21-40 yaş aralığında yaş artışının yanı sıra trendin azalmak yerine artış gösterdiği görülmektedir. 21-40 yaş grupları için model çıktıları, genel kanıya uygun sonuçlar vermemektedir; bu durum, alınan örneklemin her zaman ürünün genel öngörüsünü karşılamayabileceğini göstermektedir. Bu bağlamda, aktüer ticari olarak katsayıları kullanma kararı verebilir. Tablo 6'da, cinsiyet ayrımına dayanan model çıktıları incelendiğinde, kadın sigortalılara ait katsayının erkeklere göre yaklaşık %32 daha yüksek olduğu tespit edilmiştir. Genel trafik portföyü ile karşılaştırıldığında, kadınların hasar sıklığının erkeklere kıyasla daha yüksek olduğu, diğer trafik kullanım tarzlarında da gözlemlenmektedir. Bu bulgu, modelin genel trafik portföyü ile benzer bir durum sergilediğini göstermektedir. Tablo 6'da, model çıktıları bölgesel bazda incelendiğinde, en yüksek katsayı tahmininin ciddi bir farkla Marmara Bölgesi'ne ait olduğu görülmektedir. Marmara Bölgesi'nde, özellikle İstanbul'un büyük ticaret hacmi göz önünde bulundurulduğunda, ticari amaçla kullanılan kamyonların

trafikte daha aktif olması ve bu nedenle daha fazla kaza yaşanması, bu bölgenin diğer bölgelere göre daha yüksek katsayı almasını açıklamaktadır. Öte yandan, Güneydoğu ve Doğu Anadolu bölgelerine ait katsayıların diğer bölgelere göre daha düşük olduğu dikkat çekmektedir. Bu bölgelerdeki poliçe adetlerinin az olması, bu katsayıların güvenilirliği konusunda aktüerlerde bir soru işareti oluşturabilir; ayrıca, bu bölgelerdeki trafik yoğunluğunun da düşük olması bu sonucu destekleyebilir.

Tüm bu sonuçların ticari tarifeye geçişinde, aktüerin veriyi iyi tanıması ve modeli veriye göre doğru yorumlayabilmesi son derece önemlidir. Alınan örneklemin, her zaman portföyün genelini sağlıklı bir şekilde yansıtmayabileceği unutulmamalıdır. Bu nedenle, aktüerin elde ettiği model çıktılarını ticari tarifeye geçerken yorumlayarak aktarması ve portföy bilgisi bu süreçte kritik bir rol oynamaktadır. Sonuç olarak, hasar sıklığına ilişkin tarife denklemi (3) eşitliğindeki gibi elde edilmiştir.

$$\hat{Y} = exp\{-2{,}62387 + \hat{\beta}_1 Yaş + \hat{\beta}_2 Cinsiyet + \hat{\beta}_3 Bölge\} \quad (3)$$

Tablo 7. Kamyon veri seti için hasar tutarına ilişkin GLM sonuçları

	Tahmin	Katsayılar
BAZ	8,65060	5713,46
0-20	0,00000	1,00000
21-30	-0,69359	0,49978
31-40	-0,60986	0,54343
41-50	-0,60081	0,54837
51-60	-0,57620	0,56203
61-70	-0,54563	0,57948
70+	-0,75548	0,46979
K	-0,05216	0,94918
E	0,00000	1,00000
Akdeniz	0,00000	1,00000
İç Anadolu Bölgesi	0,00467	1,00468
Ege Bölgesi	-0,17265	0,84143
Doğu Anadolu Bölgesi	-0,16766	0,84564
Güneydoğu Anadolu Bölgesi	0,58864	1,80154
Karadeniz Bölgesi	-0,14375	0,86610
Marmara Bölgesi	-0,22223	0,80073

Trafik portföyünde genellikle yaşın artması durumunda hasar sıklığının azalması, hasar tutarının ise artması beklenir. Tablo 7'de, model çıktıları yaş değişkeni için incelendiğinde, 0-20 yaş grubuna ait tahminin diğer

gruplara göre yüksek olduğu gözlemlenmektedir. Bunun nedeni, bu yaş grubuna ait poliçe sayısının azlığı ile birlikte, gerçekleşen hasar tutarlarının yüksek olmasıdır. 20 yaş ve üzerindeki gruplara ait katsayılar ise trafik portföyünün genel eğilimiyle paralel bir trend göstermektedir; yani yaş arttıkça hasar tutarına ait katsayı artmaktadır, bu da beklenen bir durumdur. Son yaş grubunda katsayıdaki düşüşün sebebini incelemek amacıyla veri kontrol edildiğinde, bu gruba ait poliçe sayısının az olduğu ve hasarların ortalama seviyelerde gerçekleştiği sonucuna ulaşılmıştır. Tablo 7'de, model çıktıları cinsiyet değişkeni açısından değerlendirildiğinde, kadınların hasar tutarının erkeklere göre yaklaşık %5 gibi küçük bir oranla daha düşük olduğu tespit edilmiştir. Tablo 6'da, hasar sıklığı için en yüksek katsayının Marmara Bölgesi'nde olduğu belirlenmiş iken, Tablo 7'de hasar tutarında en düşük katsayıya sahip bölge, yine Marmara bölgesi olarak gözlemlenmektedir. Bu durum, trafik yoğunluğunun fazla olduğu bu bölgede daha sık kaza yapılmasına, ancak meydana gelen kazaların yoğun trafik nedeniyle düşük maliyetli olmasına işaret etmektedir. Böylece, hasar tutarına ilişkin tarife denklemi (4) eşitliğindeki gibi elde edilmiştir.

$$\hat{Y} = exp\{\, 8{,}65060 + \hat{\beta}_1 Yaş + \hat{\beta}_2 Cinsiyet + \hat{\beta}_3 Bölge \,\} \qquad (4)$$

6. Sonuçlar

Sigortacılık sektörü, ülkemizde ve dünya genelinde gelişmiş ve gelişmekte olan sektörler arasında yer almıştır. İnsanlar, karşılaşabilecekleri riskleri sigortalayarak kendileri için bir güvence oluşturmak istemişlerdir; bu nedenle sektöre yoğun bir ihtiyaç duyulmuştur. Şirketlerin, bu belirsizlik içeren riskleri karşılamak amacıyla sigortalı ile anlaşırken diğer ticari kuruluşlar gibi kendi karlılığını da sürdürebilmeleri öncelikli hedefleri arasında yer almıştır. Prim, sigortalının gerçekleştirebileceği hasar adedi ve hasar tutarı ile doğrudan ilişkilidir. Prim belirlenirken, sigorta şirketlerinin hangi prim hesaplama ilkelerini benimseyeceği, kendi ticari stratejilerine bağlı olarak değişmektedir. Prim hesabına giden bu süreçte, riskin belirsizlik içermesi nedeniyle, mevcut veriler detaylı istatistiksel teknikler ile incelenmiştir. Analize başlamadan önce veriyi iyi tanımak ve portföyü iyi analiz etmek, istatistiksel açıdan model çıktılarını yorumlama konusunda daha donanımlı tespitler yapılmasına olanak sağlamaktadır. Bu nedenle, hasar tutarı ve hasar sıklığına ait tanımlayıcı istatistiklerin elde edilmesi önemlidir. Veriye ait tanımlayıcı istatistikleri yorumladıktan sonra, maksimum ve minimum değerlerin veriyi saptırıcı bir etkisi olup olmadığı gözlemlenmiş, bu nedenle analiz, sahip olunan tüm veriler ile gerçekleştirilmiştir. Veri analiz edildikten sonra, risk modelinin bireysel mi yoksa kolektif mi oluşturulacağına karar verilmiştir. Bu çalışmada veriler bireysel bazda derecelendirmeye daha uygun olduğundan, hasar tutarı ve hasar sıklığı modeli bireysel risk modeli kullanılarak elde edilmiştir. Modele başlamadan önce hasar tutarının ve hasar sıklığının hangi dağılımlara uygun olduğu AIC ve BIC bilgi kriterlerinden

yararlanılarak, dağılım parametreleri ise EÇO yöntemi kullanılarak belirlenmiştir. Uygun dağılımın belirlenmesinin ardından GLM aşamasına geçilmiştir. GLM, sigortalıları veriye konu olan tüm kategoriler ile değerlendirebilmektedir. Bu çalışmada kategoriler yaş, cinsiyet ve coğrafi bölge değişkenleridir. Bu değişkenler için, hem hasar sıklığı hem de hasar tutarı için model çıktısındaki katsayılar ile tarife denklemleri elde edilmiştir. Elde edilen bu denklemler sonucunda sigortalının öngörülen net risk primi hesaplanmıştır. Tarife yöneticileri, portföy deneyimleri ışığında şirketin büyüme stratejisi ve ticari karlılık hedeflerini baz alarak, net risk primi üzerine gerekli yüklemeleri yapabilmekte ve sigortalıdan talep edilmesi gereken nihai primi elde edebilmektedirler.

Referanslar

- Bain, L.J. and Engelhardt, M. (1992). Introduction to Probability and Mathematical Statistics, Second Edition, Duxbury Thomson Learning, United States.
- Bahnemann, D. (2015). Distributions for actuaries. CAS monograph series, 2, 1-200.
- Denuit, M., Dhaene, J., Goovaerts, M., & Kaas, R. (2005). Actuarial theory for dependent risks: measures, orders and models. John Wiley & Sons.
- Frees, E. W., & Huang, F. (2023). The discriminating (pricing) actuary. North American Actuarial Journal, 27(1), 2-24.
- Gültekin, Ö.C., & Erdemir, C. (2010). Türkiye demir ve çelik sektöründe bir şirketin yangın risklerinin aktüeryal modeli. İstatistikçiler Dergisi: İstatistik ve Aktüerya, 3(1), 37-44.
- Kaas, R., Goovaerts, M.J., Dhaene, J., and Denuit, M. (2001). Modern Actuarial Risk Theory. Kluwer Academic Publishers, Dordrecht.
- Klugman, S. A., Panjer, H. H., & Willmot, G. E. (2012). Loss models: from data to decisions (Vol. 715). John Wiley & Sons.
- Mildenhall, S. (1999). A systematic relationship between minimum bias and generalized linear models. In Proceedings of the Casualty Actuarial Society, Vol. 86, No. 164, pp.393-487.
- Nelder, J., Wedderburn, R. (1972). Generalized Linear Models, Journal of the Royal Statistical Society, 370-384.
- Ohlsson, E., & Johansson, B. (2010). Non-life insurance pricing with generalized linear models (Vol. 174). Berlin: Springer.
- Parodi, P. (2023). Pricing in general insurance. Chapman and Hall CRC press.
- Selimović, J. (2010). Actuarial estimation of technical provisions'adequacy in life insurance companies. Interdisciplinary Management Research, 6, 523-533.
- Szegö, G.P. (Ed.). (2004). Risk measures for the 21st century (Vol. 1). New York: Wiley.
- Şahin, Ş., Karabey, U., Karageyik, B. B., Nevruz, E., & Yıldırak, K. (2016). Türkiye'de Buğday Bitkisel Ürün Sigortası için Aktüeryal Prim Hesabı. Tarım Ekonomisi Dergisi, 22(2), 37-47.

- Tse, Y.K. (2009). Nonlife actuarial models: theory, methods and evaluation. Cambridge University Press.

Bölüm 6

YAPAY SİNİR AĞLARININ SÜRÜ ZEKASI ALGORİTMALARI KULLANILARAK OPTİMİZASYONU ÜZERİNE İNCELEME

Çağatay BAL[1]

1 Arş. Gör. Dr. Muğla Sıtkı Koçman Üniversitesi, Fen Fakültesi, İstatistik Bölümü, Muğla/Türkiye, cagataybal@mu.edu.tr , https://orcid.org/0000-0002-7823-2712

1. Giriş

Yapay Sinir Ağları (YSA), belirli verilen bilgilere dayanarak çıkarımlar üretmek üzere insan beynini oluşturan biyolojik sinir ağını simüle eden hesaplamalı modellerdir. Çok sayıda alandaki sayısız sınıflandırma, regresyon, kümeleme ve ilişkilendirme problemini çözmek için hem denetimli hem de denetimsiz öğrenme için uygundurlar. Özellikle YSA, makine öğrenmesi alanında öne çıkan bir algoritma olmuş ve doğal dil işleme, sahtecilik tespiti, hesaplamalı biyoloji, bilgisayarlı görü, araçların yardımsız kontrolü, konuşma tanıma, tıbbi teşhis ve öneri sistemleri gibi birçok alandaki ilerlemenin yolunu açmıştır [1].

Son zamanlarda YSA'lar, sağlık kuruluşlarında karar vermek [2], binalardaki enerji kullanımını tahmin etmek [3], sera teknolojisinin gelişimi [4], fotovoltaik teknolojisindeki hataları tespit etmek [5] ve güneş enerjisi tahminleri [6] için uygulanmıştır.

Biyolojik sinir ağına benzer şekilde, YSA'lar, ağın diğer katmanlarındaki düğümlere ağırlıklı bağlantılarla bağlanan, düğüm (node) olarak da adlandırılan nöronlardan oluşur. Genellikle bir girdi katmanı, bir veya birkaç gizli katman ve bir çıktı katmanından oluşurlar. Ağ yapısı, yani gizli katmanların sayısı ve katmanlardaki nöronların sayısı, genellikle deneme-yanılma yöntemi kullanılarak belirlenir. Ancak, bir sinir ağının optimal yapısını akıllı bir şekilde seçmek için optimizasyon algoritmaları kullanılabilir.

Bağlantı ağırlıklarına gelince, bunlar YSA'yı kullanmadan önce gerekli bir adım olan yapay sinir ağının eğitimi sırasında belirlenir. Eğitim sürecinde, bağlantı ağırlıkları ve bias değerleri, sinir ağının verilen girdilere dayanarak doğru çıktıyı üretmesini sağlamak için ayarlanır. Denetimli öğrenme için, bir YSA'nın eğitimi, YSA'dan tahmin edilen çıktı ile eğitim verilerinden gelen hedef çıktı arasındaki hata farkının azaltılmasıyla yapılır. Dolayısıyla, yapay sinir ağları, eğitim sürecinde bağlantı ağırlıkları ve bias değerleri optimize edilerek veya ağ yapısı optimize edilerek optimize edilebilir.

Yapay sinir ağları geleneksel olarak geri yayılım (back-propagation), Quasi-Newton, eşlenik gradyan (conjugate gradient), Levenberg-Marquardt ve Gauss-Newton gibi gradyan iniş tabanlı algoritmalar kullanılarak eğitilir [7]. Bunlar yüksek sömürü (exploitation) yeteneklerine sahip yerel arama algoritmalarıdır. Ancak, keşif (exploration) yeteneklerine sahip olmadıkları için, genellikle arama uzayında YSA için optimal bağlantı ağırlıklarını bulmakta yetersiz kalırlar ve sıklıkla yerel optimumlara takılırlar [7]. Bu durum, eğitilen sinir ağlarının düşük bir doğruluğa sahip olmasına neden olur.

Son zamanlarda, sinir ağlarının eğitimi için sürü zekası algoritmalarının kullanımının, keşif ve sömürü yetenekleri sayesinde daha faydalı olduğu fark edilmiştir. Ayrıca, ağ yapısını akıllı bir şekilde belirlemek için de yararlıdırlar.

Sürü zekası algoritmaları, hayvan gruplarının veya böcek sürülerinin kendi aralarında ve çevreleriyle etkileşime girerken sergiledikleri davranışlardan esinlenen meta-sezgisel optimizasyon algoritmalarıdır. Özellikle, küresel bir zeka ortaya çıkaran bir biyolojik organizma popülasyonunun basit kolektif davranışını kullanırlar. Bu, sürü zekası algoritmalarının, kendi aralarında ve çevreleriyle etkileşime giren yapay arama ajanlarından oluşan bir popülasyon kullanarak karmaşık optimizasyon problemlerini çözmelerine olanak tanır.

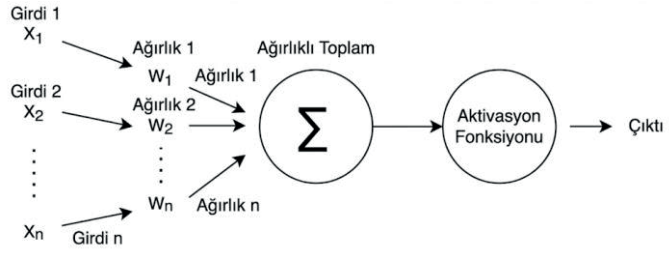
Yapay sinir ağlarını optimize etmek için çok sayıda sürü zekası algoritması ve bunların hibritleri kullanılmıştır. Ancak, bildiğimiz kadarıyla, yapay sinir ağlarını optimize etmek için kullanılan farklı sürü zekası algoritmaları üzerine kapsamlı bir inceleme bulunmamaktadır.

Bu makalede, sürü zekası algoritmaları kullanılarak optimize edilen farklı YSA türleri, YSA'ların sürü zekası algoritmaları kullanılarak optimize edilme yolları, kullanılan farklı orijinal ve hibrit algoritmalar ve eğitilen sinir ağlarının kullanıldığı farklı denetimli öğrenme görevleri açısından, YSA'ları optimize etmek için kullanılan sürü zekası algoritmalarının bir incelemesini sunuyoruz.

2. Sinir Ağları Ve Sürü Zeka Algoritmaları Metodolojileri

2.1. Yapay Sinir Ağları

En basit YSA, girdi değerleri, ağırlıklar, biaslar, net toplam ve bir aktivasyon fonksiyonundan oluşan tek katmanlı bir sinir ağı olan algılayıcıdır (perceptron). Genellikle denetimli öğrenmede girdileri iki sınıftan birine kategorize etmek için ikili sınıflandırıcı olarak kullanılır. Ağırlıklar her bir girdinin önemini temsil eder ve girdilerin ağırlıklı toplamı, verileri sınıflandırmak için aktivasyon fonksiyonu tarafından kullanılır. Şekil 1, bir algılayıcının yapısını göstermektedir.



Şekil 1. Bir algılayıcının yapısı. Girdiler $X_1, X_2 \ldots X_n$, Ağırlıklar $W_1, W_2 \ldots W_n$, Ağırlıklı Toplam Σ, Aktivasyon Fonksiyonu → Çıktı

Algılayıcı sadece basit ikili problemler için kullanılabilir. Bu nedenle, algılayıcı temel yapı taşı olarak alınarak çok sayıda başka sinir ağı geliştirilmiştir. Bu genellikle, her katmanda bir veya daha fazla algılayıcı bulunan çoklu algılayıcı katmanları kullanılarak yürütülür. Düğüm veya nöron olarak da adlandırılan her katmandaki algılayıcılar, bir önceki katmandakiler tarafından elde edilen sonuçlara dayanarak çıkarımlar yapar ve sonuçları bir sonraki katmandaki algılayıcılara iletir.

Üç ana YSA türü vardır; İleri Beslemeli Sinir Ağları (FNN - Feedforward Neural Networks), Yinelemeli sinir Ağları (RNN - Recurrent Neural Networks) ve Evrişimsel Sinir Ağları (CNN - Convolutional Neural Networks). İleri beslemeli sinir ağları, bilginin sadece girdi düğümlerinden gizli katman düğümleri aracılığıyla ve son olarak çıktı düğümlerine doğru tek yönde aktarıldığı YSA'lardır. Herhangi bir bağlantı döngüsü içermezler ve genellikle bağımsız değişkenler arasındaki örüntüleri tanımlamak için kullanılırlar.

Yinelemeli sinir ağları, bağlantı döngülerinin olduğu, yani bir katmanın çıktısının çıktıyı belirlemek için girdiye geri iletildiği YSA'lardır. Bu, çıktıyı üretmek için tüm bilgileri saklayan bir hafıza gibi davranır. RNN'ler genellikle bir dizi içeren, yani birbirine bağımlı değişkenler veya zaman serisi verileri için kullanılır.

Evrişimsel sinir ağları, esas olarak iki boyutlu görüntülerle çalışmak için tasarlanmış özelleşmiş sinir ağlarıdır. Genellikle verilerdeki özellikleri tanımlayan evrişim katmanları (convolution layers), havuzlama katmanları (pooling layers) ve tam bağlantılı katmanlar (fully connected layers) gibi çeşitli bloklardan oluşurlar.

2.2 Sürü Zekası Algoritmaları

Sürü zekası algoritmaları, doğadaki çeşitli biyolojik organizma popülasyonlarından esinlenmiştir. Genellikle belirli görevleri basit organize adımlar kullanarak akıllıca başarmak için kendi aralarında ve çevreleriyle etkileşime giren belirli organizmaların belirli özelliklerini kopyalarlar. Sürü zekası algoritmaları, bir arama uzayında belirli bir biyolojik organizma grubuna benzer şekilde etkileşime giren yapay arama ajanları popülasyonundan oluşur. Bu basit etkileşimler, algoritmanın bir problem için optimal çözümü sezgisel bir şekilde aramasını sağlar. Dolayısıyla, makul bir zaman diliminde optimal veya ideale yakın çözümler sağlayarak sayısız optimizasyon problemini çözebilirler.

Sürü zekası algoritmaları, sürekli, ayrık veya çok amaçlı optimizasyon problemleri dahil olmak üzere farklı türdeki optimizasyon problemlerini çözmek için kullanılabilir. Bu nedenle, çeşitli alanlarda çok sayıda uygulamaları vardır. Örneğin, su kaynakları mühendisliğinde [8], kablosuz ağlarda [9], bulut tabanlı Nesnelerin İnterneti'nde [10], optik sistemlerde [11], öneri sistemlerinde [12], anomali tespit sistemlerinde [13] ve tedarik zinciri yönetiminde [14] kullanılabilirler. Ayrıca kümeleme [15], öznitelik seçimi [16] ve gezgin satıcı problemini çözmek [17], [18] için de kullanılabilirler. Dahası, optimal tasarımlar, elektrik mühendisliği, ağ oluşturma, makine mühendisliği, makine öğrenmesi, kaynak tahsisi ve dijital görüntü işleme alanlarında çeşitli uygulamaları vardır [19].

2.3. Farklı Türdeki Yapay Sinir Ağlarının Optimizasyonu

Çeşitli uygulamalar için kullanılan farklı mimarilere sahip birçok farklı türde yapay sinir ağı mevcuttur. Sinir ağlarının çoğu, üç ana YSA türünden biri olarak sınıflandırılabilir: ileri beslemeli sinir ağı, yinelemeli sinir ağı veya evrişimsel sinir ağı.

Sürü zekası algoritmaları kullanılarak optimize edilmiş dört tür ileri beslemeli sinir ağı olduğunu bulduk; Çok Katmanlı Algılayıcı (MLP), Derin İleri Beslemeli ağ, Aşırı Öğrenme Makinesi (ELM) ve Radyal Tabanlı Fonksiyon (RBF) sinir ağı.

Sürü zekası algoritmaları tarafından optimize edilen beş tür YSA, yinelemeli sinir ağları olarak sınıflandırılabilir: Dışsal girdili Doğrusal Olmayan Otoregresif (NARX), Elman RNN, derin RNN, Uzun Kısa Süreli Bellek (LSTM) ve Jordan RNN.

Ayrıca, evrişimsel sinir ağları olarak sınıflandırılabilen ve sürü zekası algoritmaları kullanılarak optimize edilmiş dört tür YSA vardır; Görsel Geometri Grubu Sinir Ağı (VGGNet), Kalıntı Sinir Ağı (ResNet), GoogLeNet ve U-Net.

İlgili literatürdeki çalışmalardan; [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45] ve [46]'da sürü zekası algoritmaları ileri beslemeli çok katmanlı algılayıcıları optimize etmek için kullanılmıştır. Çok katmanlı bir algılayıcı, birden fazla algılayıcıdan oluşan bir sinir ağıdır. Genellikle bir girdi katmanı, değişken sayıda gizli katman ve bir çıktı katmanından oluşur ve her biri değişen sayıda algılayıcıya sahiptir. Her katmanın algılayıcıları, önceki ve sonraki katmanların algılayıcılarına bağlantı ağırlıkları aracılığıyla bağlanır.

Çalışma [20]'de, girdi katmanında 3, 4, 4, 9, 9 ve 22 nöron, gizli katmanda 7, 9, 9, 19, 19 ve 45 nöron ve çıktı katmanında 1, 3, 1, 1, 1, 1 nöron içeren altı MLP kullanılmıştır. [21]'de, girdi, gizli ve çıktı katmanlarında sırasıyla yedi, dört ve bir düğümden oluşan üç MLP kullanılmıştır. [22]'de, girdi, gizli ve çıktı katmanlarında sırasıyla 20, 50 ve bir nöron içeren beş MLP kullanılmıştır. [23]'te, girdi, gizli ve çıktı katmanlarında sırasıyla beş, 10 ve bir nöron içeren bir MLP kullanılmıştır. [24]'te, girdi, gizli ve çıktı katmanlarında sırasıyla altı, on ve bir düğüm içeren bir MLP kullanılmıştır. [25]'te, kullanılan MLP girdi, gizli ve çıktı katmanlarında sırasıyla iki, üç ve bir nörondan oluşmaktadır. [26]'da, girdi, gizli ve çıktı katmanlarında sırasıyla yedi, altı ve bir düğüm içeren iki MLP kullanılmıştır. [27]'de, sırasıyla 3-7-1, 4-9-1 ve 22-45-1 yapılarına sahip üç MLP kullanılmıştır. [28]'de, 8-9-1 yapısına sahip bir MLP kullanılmıştır. [29]'da, 12-8-1 yapısına sahip bir MLP kullanılmıştır. [31]'de, 14-5-5 yapısına sahip bir MLP kullanılmıştır. [32]'de, 4-7-1 yapısına sahip bir MLP kullanılmıştır. [33]'te, 2-12-1 yapısına sahip bir MLP kullanılmıştır. [35]'te, 8-1-1, 8-2-1, 8-3-1, 8-4-1 ve 8-5-1 yapılarına sahip beş MLP kullanılmıştır. [36]'da, 20-10-1 yapısına sahip bir MLP kullanılmıştır. [37]'de, 32-64-6 yapısına sahip bir MLP kullanılmıştır. [40]'ta, 7-5-1 yapısına sahip bir MLP kullanılmıştır. [41]'de, 14-11-18 yapısına sahip bir MLP kullanılmıştır. [43]'te, 3-7-1, 4-9-1, 4-9-3, 13-27-3, 9-19-1, 22-45-1, 4-9-1, 6-13-1, 10-21-1, 14-29-1, 1-15-1, 1-15-1 ve 1-15-1 yapılarına sahip 13 MLP kullanılmıştır. [44]'te, 9-19-1, 9-18-1 ve 9-17-1 yapılarına sahip üç MLP kullanılmıştır. [46]'da, 4-3-3, 4-3-1, 9-3-1, 9-3-1, 4-9-3, 4-9-1, 9-19-1 ve 9-19-1 yapılarına sahip 8 MLP kullanılmıştır.

[47], [48], [49] ve [50]'de sürü zekası algoritmaları derin ileri beslemeli sinir ağlarını optimize etmek için kullanılmıştır. Derin ileri beslemeli ağlar, birden fazla gizli katmana sahip ileri beslemeli sinir ağlarıdır. [48]'de, kullanılan derin sinir ağının mimarisi, 13 nöronlu bir girdi katmanı, her biri altı nöronlu 4 gizli katman ve bir nöronlu bir çıktı katmanından oluşmaktadır.

[51]'de, bir sürü zekası algoritması tarafından eğitilmek üzere bir aşırı öğrenme makinesi (ELM) kullanılmıştır. Bir ELM, tam olarak bir gizli katmandan oluşan ileri beslemeli bir sinir ağıdır ve girdi ile gizli düğümler arasındaki ağırlıklar rastgele atanır ve değişmezler. Ağın eğitimi sırasında, gizli düğümler ile çıktı düğümleri arasındaki ağırlıklar optimize edilir. [52]'de, bir sürü zekası algoritması, bir girdi, bir gizli ve bir çıktı katmanından oluşan ve aktivasyon fonksiyonları olarak Gauss fonksiyonlarını kullanan ileri beslemeli bir sinir ağı olan radyal tabanlı fonksiyon (RBF) sinir ağını eğitmek için kullanılmıştır.

[53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64] ve [65]'te, sürü zekası algoritmaları farklı türdeki yinelemeli sinir ağlarını optimize etmek için kullanılmıştır. [53]'te, Dışsal Girdili Doğrusal Olmayan Otoregresif (NARX) yinelemeli sinir ağı kullanılmıştır. Diğer yinelemeli sinir ağlarının aksine, NARX'ta geri besleme, gizli nöronlar gibi diğer tüm nöronlardan gelmek yerine sadece çıktı nöronundan gelir. Kullanılan NARX sinir ağı, yedi nörona sahip bir gizli katmana sahiptir.

[54], [55], [56] ve [57]'de Elman yinelemeli sinir ağları sürü zekası algoritmaları kullanılarak optimize edilmiştir. Elman RNN'leri, gizli katmana bağlı bağlam birimi (context unit) adı verilen ek bir birime sahip yinelemeli sinir ağlarıdır. [54]'te, kullanılan Elman RNN, sırasıyla üç, altı ve bir nöronlu bir girdi, bir gizli ve bir çıktı katmanına sahiptir. [55]'te, kullanılan Elman sinir ağı, beş nöronlu bir girdi katmanı, sırasıyla 15 ve 20 nöronlu iki gizli katman ve bir nöronlu bir çıktı katmanından oluşmaktadır. [56]'da, kullanılan RNN, her katmanda değişken sayıda nöron bulunan üç gizli katmandan oluşmaktadır. [57]'de, kullanılan Elman RNN, beş düğümlü bir gizli katmandan oluşmaktadır. [58] ve [59]'da, birden fazla gizli katmana sahip derin yinelemeli sinir ağları kullanılmıştır.

[60], [61], [62], [63] ve [64]'te Uzun Kısa Süreli Bellek (LSTM) ağı kullanılmıştır. LSTM, bilgiyi uzun bir süre boyunca saklayan bellek hücresi adı verilen ek bir birime sahip tekrarlayan bir sinir ağıdır. Bu, katmanlara giren girdiyi ve ağın unutması gereken bilgileri düzenleyerek ağın daha uzun vadeli bağımlılıkları öğrenmesini sağlar. [65]'te, bir Jordan RNN'i eğitmek için bir sürü zekası algoritması kullanılmıştır. Bir Jordan RNN, ağın çıktısından gizli katmanlara bir bağlantı döngüsüne sahip olan bir RNN'dir.

[66], [67] ve [68]'de, farklı türdeki evrişimsel sinir ağları sürü zekası algoritmaları kullanılarak optimize edilmiştir. [66]'da, Görsel Geometri Grubu Sinir Ağı (VGGNet), Kalıntı Sinir Ağı (ResNet) ve GoogLeNet olmak üzere üç tür evrişimsel sinir ağı kullanılmıştır. VGGNet, 16 veya 19 katmana sahip bir evrişimsel sinir ağıdır. Evrişim katmanlarını takip eden bir havuzlama katmanından oluşur ve yaklaşık 64 ila 512 filtre kullanır. Evrişim çekirdekleri

(kernels) ve maksimum havuzlama çekirdekleri sırasıyla 3×3 ve 2×2 sabit boyutuna sahiptir. ResNet, 50, 101 veya 152 katmandan oluşan ve atlama bağlantıları (skip connections) adı verilen kapılı tekrarlayan birimleri içeren bir CNN'dir. GoogLeNet, 22 katman ve 9 başlangıç modülünden (inception modules) oluşur. Başlangıç modülleri, ağın her blokta sabit bir boyut kullanmak yerine evrişimsel filtrenin boyutunu seçmesine izin verir. [68]'de, 16 ve 19 katmanlı VGGNet kullanılmıştır. [67]'de, bir U-Net CNN kullanılmıştır. U-Net, U şeklinde bir kodlayıcı-kod çözücü ağ mimarisine sahip tam bağlantılı bir CNN'dir. Bir köprü ile bağlanan dört kodlayıcı bloğu ve dört kod çözücü bloğu vardır.

Yukarıdaki açıklamalara ve taksonomiye dayanarak, çoğunlukla ileri beslemeli sinir ağlarının, özellikle MLP'lerin sürü zekası algoritmaları kullanılarak optimize edildiği görülebilir. RNN'ler arasında, çoğunlukla Elman RNN'leri ve LSTM'ler sürü zekası algoritmaları kullanılarak optimize edilmiştir. CNN, sürü zekası algoritmaları tarafından en az optimize edilen sinir ağı türüdür. Sadece beş CNN sürü zekası algoritmaları kullanılarak optimize edilmiştir.

2.4 Yapay Sinir Ağı Parametre ve Mimari Optimizasyonu

Sürü zekası algoritmaları, yapay sinir ağlarının performansını, yalnızca bağlantı ağırlıklarını optimize ederek, ağırlıkları ve biasları optimize ederek veya ağ yapısını optimize ederek optimize edebilir.

[22] ve [24]'te, sürü zekası algoritmaları sadece ileri beslemeli sinir ağlarının bağlantı ağırlıklarını optimize etmek için kullanılmıştır. İleri beslemeli sinir ağlarında, bir katmandaki nöronlar, önceki ve sonraki katmanların nöronlarına, bağlantının gücünü gösteren ağırlıklar kullanılarak bağlanır. Ağın eğitimi sırasında, bu ağırlıklar, ağın tahmin edilen çıktısı ile hedef çıktı arasındaki hatanın azaltılacağı şekilde uyarlanır. Bu nedenle, hatayı amaç fonksiyonu olarak alarak, sürü zekası algoritmaları eğitim sürecinde sinir ağının optimal bağlantı ağırlıklarını bulmak için uygulanır.

İleri beslemeli sinir ağlarının eğitimi sırasında, bağlantı ağırlıklarının yanı sıra, sinir ağının çıktıyı ayarlayarak veya aktivasyon fonksiyonunu kaydırarak çıktıyı en iyi şekilde tahmin etmesini sağlamak için bias (sapma) adı verilen sabitler de belirlenir. Bu biaslar da sinir ağının hatasını amaç fonksiyonu olarak alarak optimize edilir. [20], [21], [23], [25], [26], [27], [28], [29], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [43], [44], [45], [46], [51] ve [52]'de, sürü zekası algoritmaları ileri beslemeli sinir ağlarının bağlantı ağırlıklarını ve biaslarını optimize etmek için kullanılır.

Sinir ağlarının yapısı, yani katman sayısı, aktivasyon fonksiyonu, her katmandaki nöron sayısı ve öğrenme oranı, genellikle ağın eğitiminden önce deneme-yanılma temelinde belirlenir. Ancak, optimizasyon algoritmaları kullanılarak akıllı bir şekilde de belirlenebilirler. [30], [36], [42], [49] ve [50]'de, sürü zekası algoritmaları ileri beslemeli sinir ağlarının yapısını optimize etmek için kullanılmıştır. [30]'da, bir sürü zekası algoritması, bir ileri beslemeli sinir ağının gizli katman sayısını, gizli katmandaki nöron sayısını, aktivasyon fonksiyonunu ve öğrenme oranını optimize etmek için kullanılmıştır. [36]'da, ağırlıklar ve biaslara ek olarak, gizli katmandaki nöron sayısı da bir sürü zekası algoritması tarafından optimize edilir. [49]'da, epoch sayısı, öğrenme oranı, parti (batch) boyutu ve ileri beslemeli bir sinir ağının ilk gizli katmanındaki nöron sayısı bir sürü zekası algoritması tarafından optimize edilir. [42]'de, gizli katman sayısı, gizli katmandaki nöron sayısı ve öğrenme oranı bir sürü zekası algoritması tarafından optimize edilir. [50]'de, bir ileri beslemeli sinir ağının gizli katman sayısı bir sürü zekası algoritması kullanılarak optimize edilir.

[53], [54] ve [58]'de, yinelemeli sinir ağlarının bağlantı ağırlıkları sürü zekası algoritmaları kullanılarak optimize edilir. [56], [57], [58], [59], [63], [65] ve [69]'da, eğitim sırasında yinelemeli sinir ağlarının hem bağlantı ağırlıkları hem de biasları optimize edilir. [60], [61], [62] ve [64]'te, yinelemeli sinir ağlarının optimal yapısı, eğitimlerinden önce sürü zekası algoritmaları kullanılarak belirlenir. [61]'de, bir yinelemeli sinir ağının gizli katman sayısı, zaman gecikmeleri, parti boyutu, gizli katmanlardaki nöron sayısı ve epoch sayısı bir sürü zekası algoritması tarafından optimize edilir. [62]'de, tekrarlayan katmandaki nöron sayısı, çıktı dropout (sönümleme), tekrarlayan dropout değeri ve gömme (embedding) dropout değeri bir sürü zekası algoritması kullanılarak optimize edilir. [64]'te, bir yinelemeli sinir ağının parti boyutu, zaman adımı ve gizli katman birimlerinin sayısı bir sürü zekası algoritması kullanılarak optimize edilir.

[67]'de, bir evrişimsel sinir ağının bağlantı ağırlıkları bir sürü zekası algoritması tarafından optimize edilir. [70]'te, bir evrişimsel sinir ağının optimal bağlantı ağırlıkları ve biasları bir sürü zekası algoritması tarafından belirlenir. [66], [68], [71] ve [72]'de, evrişimsel sinir ağlarının yapısı sürü zekası algoritmaları kullanılarak optimize edilir.

Yukarıdaki açıklamalara ve taksonomiye dayanarak, sürü zekası algoritmalarının öncelikle ağırlıkları ve biasları optimize etmek, yani ileri beslemeli sinir ağlarını eğitmek için kullanıldığı çıkarımı yapılabilir. Ayrıca, yinelemeli sinir ağlarının eğitimi için de önemli ölçüde kullanılmışlardır. Ancak, ağ yapısını optimize etme açısından, sürü zekası algoritmaları hem FNN'ler hem de RNN'ler için nadiren kullanılmıştır. CNN'lere gelince, ağ yapısını optimize etmek için birkaç sürü zekası algoritması kullanılmıştır.

Ancak, sadece iki çalışmada CNN'lerin ağırlıkları ve biasları sürü zekası algoritmaları kullanılarak optimize edilmiştir.

3. Sinir Ağları Optimizasyonu İçin Kullanılan Sürü Zekası Algoritmaları

Son yıllarda, farklı biyolojik organizmalardan esinlenen çok sayıda sürü zekası algoritması önerilmiştir. Bunların birçoğu farklı türdeki sinir ağlarını optimize etmek için uygulanmıştır. Bu bölümde, ileri beslemeli, tekrarlayan ve evrişimsel sinir ağlarını optimize etmek için kullanılan farklı sürü zekası algoritmalarının bir açıklaması verilmektedir.

3.1. İleri Beslemeli Sinir Ağları

Literatürdeki çalışmalardan; [22], [23], [29], [32], [33], [35], [39], [40], [42], [44], [45], [48] ve [50]'de, orijinal Parçacık Sürü Optimizasyonu (PSO) algoritması YSA'ları eğitmek için kullanılmıştır. PSO, kuş sürülerinin veya balık sürülerinin hareketini simüle eden bir sürü zekası algoritmasıdır. Çözüm uzayındaki her parçacığın konumu, optimizasyon problemi için bir potansiyel çözümü temsil eder. Her parçacık konumunu kendi deneyimine ve tüm popülasyonun deneyimine göre günceller.

[24]'te, PSO'nun keşif ve sömürü aşamalarını dengeleyen yeni bir değişken atalet ağırlığına sahip hibrit bir PSO kullanılır. [30]'da, PSOCOG adı verilen, ağırlık merkezi konseptine sahip değiştirilmiş bir PSO kullanılır. PSOCOG, referans noktasından uzaklıklarına göre hesaplanan popülasyondaki parçacıkların ortalama ağırlığını kullanır. [51]'de, bir YSA'yı eğitmek için zamanla değişen hızlanma katsayılarına sahip Değiştirilmiş bir PSO (MPSO) kullanılır. [34]'te, bir YSA'yı eğitmek için Fork/Join çerçevesini içeren bir Paralel PSO (PPSO) kullanılır. [36]'da, DPSO-PSO-FFNN adı verilen ayrık PSO, PSO ve Levenberg-Marquardt algoritmalarının bir hibriti kullanılır. [38]'de, keşif ve sömürü aşamalarını dengelemek için insan tabanlı PSO'da hücresel otomat algoritmasını içeren bir PSO hibriti olan HPSO-CA algoritması kullanılır. [45]'te, bir sinir ağını eğitmek için İşbirlikçi Kuantum PSO (CQPSO) adı verilen dinamik bir PSO kullanılır.

[21], [25] ve [26]'da, ileri beslemeli sinir ağlarını eğitmek için orijinal Yusufçuk Algoritması (DA) kullanılır. DA, yusufçuk sürülerinden, özellikle avlanma ve göç etme davranışlarından esinlenmiştir. [20]'de, DA ve Yapay Arı Kolonisi'nin (ABC) bir hibriti olan HAD kullanılır. [27]'de, bir sinir ağını eğitmek için DA ve geliştirilmiş bir Nelder-Mead algoritmasının hibriti olan INMDA kullanılır. INMDA algoritmasında, geliştirilmiş Nelder-Mead

algoritması orijinal DA'nın yerel arama yeteneğini geliştirir. [46]'da, MLP'leri eğitmek için DA ve tepe tırmanma (hill climbing) algoritmasının bir hibriti kullanılır. Tepe tırmanma algoritması, orijinal DA'nın sömürü aşamasını geliştirmek için kullanılır.

[22]'de, PSO'ya ek olarak, Karınca Kolonisi Optimizasyonu (ACO), Yapay Arı Kolonisi (ABC) ve ateş böceği algoritmaları YSA'ların eğitimi için kullanılır. ACO, karınca kolonilerinin yiyecek arama davranışından, yani yollarına feromon izleri bırakarak kolonilerinden yiyecek kaynaklarına en kısa yolu bulma biçimlerinden esinlenmiştir. ABC, arıların yiyecek arama davranışından esinlenmiştir. Bir kolonide üç tür arı vardır; işçi arılar, gözcü arılar ve kaşif arılar. İşçi arılar keşfedilen yiyecek kaynaklarının sayısını ve kalitesini takip eder, gözcü arılar en iyi yiyecek kaynaklarını seçmekten sorumludur ve kaşif arılar yeni yiyecek kaynakları aramaktan sorumludur. Ateş böceği algoritması, avı veya diğer ateş böceklerini çekmek için kullanılan ateş böceği sürülerinin yanıp sönme davranışına dayanır. Ateş böcekleri, yayılan ışığın yoğunluğuna göre diğer ateş böceklerine çekilir.

[47]'de, ileri beslemeli bir sinir ağını eğitmek için ABC'nin ABC-ISB adı verilen bir hibriti kullanılır. ABC-ISB algoritması, işçi ve gözcü arılar için yinelemeli formülü değiştirerek ABC'nin yerel arama yeteneğini geliştirir. Ayrıca, kaşif arılar için yeni bir seçim stratejisi getirerek küresel arama yeteneğini de geliştirir.

[26]'da, orijinal DA'ya ek olarak, ileri beslemeli bir sinir ağının eğitimi için orijinal Harris Şahinleri Optimizasyonu (HHO) algoritması kullanılmıştır. Ayrıca [31]'de de bir ileri beslemeli sinir ağının eğitimi için kullanılmıştır. HHO algoritması, şahinlerin avı izlerken, kuşatırken, yaklaşırken ve sonunda saldırırken kullandıkları işbirlikçi avlanma stratejisinden esinlenmiştir. Keşif, şahinlerin avı tespit etmek için kullandıkları stratejilere dayanırken, sömürü avı saldırmak için kullanılan stratejilere dayanır.

[28]'de, orijinal Çekirge Optimizasyon Algoritması (GOA) ve orijinal Gri Kurt Optimizasyonu (GWO) algoritması ileri beslemeli sinir ağlarını optimize etmek için kullanılmıştır. GOA, çekirgelerin yaşam döngülerinin iki evresi olan nimf evresi ve yetişkinlik evresi sırasındaki sürü davranışına dayanır. Algoritmanın sömürü aşaması yavaş hareketi içeren nimf evresine göre modellenirken, keşif aşaması uzun ve ani hareketleri içeren yetişkinlik evresine göre modellenir. GWO, gri kurtların liderlik hiyerarşisine ve avlanma davranışına dayanır. Avı avlarken, ararken, kuşatırken ve saldırırken alfa, beta, delta ve omega olmak üzere dört tür gri kurdun davranışını simüle eder.

[52]'de, ileri beslemeli bir sinir ağını eğitmek için orijinal Rekabetçi Sürü Optimizasyonu (CSO) algoritması kullanılır. CSO, parçacık sürü

optimizasyonuna dayanır ancak sürüdeki parçacıklar arasında rekabetçi bir mekanizma kullanır. PSO'nun aksine, CSO'da parçacıkların geçmiş en iyi konumları saklanmaz ve parçacıklar rekabet kazananlarından öğrenir. [37]'de, değiştirilmiş bir Şempanze Optimizasyon Algoritması (ChOA) kullanılır. ChOA, av ararken saldırgan, sürücü, engelleyici ve kovalayıcı şempanzelere dayanır. Değiştirilmiş ChOA, dört tür şempanzenin konumlarından türetilen orantısal katsayılara dayalı olarak şempanzelerin konumlarını güncellemek için bir teknik önerir.

[49]'da, yarasa algoritmasının bir hibriti kullanılır. Yarasa algoritması, doğadaki mikro yarasaların ekolokasyonuna dayanır. Hibrit algoritma, diferansiyel evrimin mutasyon stratejilerini yarasa algoritmasına dahil eder. [41]'de, YSA eğitimi için orijinal Biyocoğrafya Tabanlı Optimizasyon (BBO) algoritması kullanılır. BBO, biyocoğrafyanın ana süreçlerinden, yani biyolojik türlerin evrimi, yok oluşu ve dağılımından esinlenmiştir. [43]'te salp sürüsü algoritmasının bir hibriti kullanılır. Salp sürüsü algoritması, salp zincirleri oluşturarak yiyecek kaynaklarına doğru hareket eden salplerin davranışından esinlenmiştir. Hibrit algoritma, keşif yeteneklerini geliştirmek için salp sürüsü algoritmasında tek bir lider yerine birden fazla liderin kullanımını içerir.

[44]'te, ileri beslemeli bir sinir ağını eğitmek için orijinal Balina Optimizasyon Algoritması (WOA) kullanılır. WOA, kambur balinaların yiyecek arama davranışından esinlenmiştir. Ana aşamalar, daralan kuşatma ve spiral güncelleme aşamalarından oluşur.

Tablo 1, kullanılan popülasyon büyüklüğü ve maksimum yineleme sayısı ve eğitilen sinir ağının nihai Kök Ortalama Kare Hatası (RMSE) açısından ileri beslemeli sinir ağlarını eğitmek için kullanılan farklı sürü zekası algoritmalarının bir karşılaştırmasını göstermektedir. Sinir ağının performansını değerlendirmek için RMSE'nin kullanılmadığı durumlarda, kullanılan performans metriği parantez içinde belirtilmiştir.

Tablo 1'den, orijinal PSO ve hibritlerinin ileri beslemeli sinir ağlarını optimize etmek için yaygın olarak kullanıldığı çıkarımı yapılabilir. 13 çalışmada orijinal PSO kullanılmış ve yedi çalışmada hibritleri kullanılmıştır. Ayrıca, genellikle daha iyi performans gösteren hibritlerinin aksine, sürü zekası algoritmalarının orijinal versiyonlarının daha fazla kullanıldığı görülebilir.

Farklı çalışmalar için farklı popülasyon büyüklükleri ve yineleme sayıları kullanılmıştır. Daha büyük popülasyon büyüklüklerinin ve daha yüksek yineleme sayılarının, optimizasyon için daha fazla zaman ve kaynağa ihtiyaç duyulduğundan düşük verimliliğe neden olabileceği not edilebilir. Bu nedenle, mümkün olan en küçük popülasyon büyüklüğü ve yineleme sayısını

kullanarak sinir ağının en düşük RMSE'sini veya en yüksek doğruluğunu elde etmek daha iyidir. Optimize edilen sinir ağlarının performansını değerlendirmek için, ağın RMSE'si en yaygın kullanılanıdır, bunu doğruluğu takip eder.

Tablo 1. İleri Beslemeli Sinir Ağlarını Eğitiminde Kullanılan Algoritmaların Karşılaştırması.

Makale	Algoritma	Popülasyon büyüklüğü	Yineleme sayısı	YSA'nın RMSE Aralığı
[22]	Orijinal PSO, ACO, ABC, Firefly	20, 100, 40, 20	100, 200, 100, 100	64.5–96.2, 34.7–96.2, 52.5–96.2, 20.7–95.6 (doğruluk)
[23]	Orijinal PSO	85	1000	14.4–2.49
[29]	Orijinal PSO	100	3000	0.0267
[32]	Orijinal PSO	25	300	0.082–0.273
[33]	Orijinal PSO		1000	0.6391
[35]	Orijinal PSO			0.515–1.264
[48]	Orijinal PSO	50	500	0.89 (AUC)
[39]	Orijinal PSO	30	50	98.0 (doğruluk)
[40]	Orijinal PSO	20	1000	0.03162
[42]	Orijinal PSO	10	500	0.1911–0.2032
[50]	Orijinal PSO	30	300	95.0 (doğruluk)
[44]	Orijinal PSO, WOA	50	500	2.456, 2.281
[45]	Orijinal PSO, Cooperative Quantum PSO (CQPSO)	10	100	2.0–3.0, 1.0–1.75 (Ortalama hata)
[24]	Hibrit PSO	50	50	97.2–99.5 (doğruluk)
[30]	PSOCOG	20	100	1.195E-05–0.225 (Ortalama Mutlak Hata)
[51]	Modified PSO (MPSO)	30	200	0.0164–0.0616
[34]	Parallel PSO (PPSO)			635.5932–1501.1942
[36]	DPSO-PSO	80	100	1.59E-02–7.78E-02
[38]	HPSO-CA	186-438	30	0.09798–0.4472
[21]	Orijinal DA			85 (doğruluk)
[25]	Orijinal DA	208	250	92.2–96.7 (doğruluk)
[26]	Orijinal DA, HHO	10-100	1000	0.3422, 0.3674
[46]	Optimized DA	05-Oct	Oct-20	0.0991–0.78259
[20]	HAD	50	100	1.292E-22–2.356E-02
[27]	INMDA		200	2.341E-16–0.3382
[47]	ABC-ISB	100	3000	4.0130E-06–0.002317
[31]	Orijinal HHO		1000	0.4444
[28]	Orijinal GOA, GWO	10-500	1000	2.4087–2.4459, 2.2899–2.2929
[52]	Orijinal CSO	30	3000	5.693E-03–8.149E-03

[37]	Modified ChOA		300	0.01414–0.5226
[49]	Hibrit Bat	40	100	96.5 (doğruluk)
[41]	Orijinal BBO	20	500	0.1584
[43]	Hibrit Salp Swarm	50, 200	250	2.381E-10–0.5617

3.2. Yinelemeli Sinir Ağları

[55] ve [61]'de, orijinal PSO yinelemeli sinir ağlarını optimize etmek için kullanılır. [54], [59], [64], [65] ve [69]'da PSO'nun hibritleri kullanılmıştır. [54]'te, küresel en iyi parçacığı seçmek için yeni bir teknik kullanan değiştirilmiş bir PSO kullanılır. [69]'da, tekrarlayan bir sinir ağını optimize etmek için BAPSO adı verilen PSO ve yarasa algoritmasının bir hibriti kullanılır. BAPSO algoritması, parçacıkların atalet ağırlığını ve ilk hızını güncellemek için yeni teknikler içerir. [59]'da, yeniden başlatma stratejisi, erken durdurma kuralı, doğrusal olarak güncellenen katsayılar ve atalet ağırlığı ve sınırlı hızlara sahip değiştirilmiş bir PSO kullanılır. [64]'te, tekrarlayan bir sinir ağını optimize etmek için uyarlanabilir bir öğrenme stratejisine sahip bir hibrit PSO kullanılır. [65]'te, tekrarlayan bir sinir ağını eğitmek için PSO-CS adı verilen PSO ve guguk kuşu (cuckoo) arama algoritmasının bir hibriti kullanılır.

[53]'te, tekrarlayan bir sinir ağını optimize etmek için orijinal Sürekli Sığırcık Sürüsü Optimizasyonu (CFSO) kullanılır. CFSO algoritması, PSO'ya dayanır ancak topolojik bir bileşene sahiptir, yani parçacıklar kişisel ve küresel en iyi konumlara çekilir ve hız bir parçacık alt kümesine göre güncellenir. [58]'de, gelişmiş sömürü ve keşif aşamalarına sahip üçlü arı (ternary bees) algoritmasının bir hibriti olan gelişmiş üçlü arı algoritması kullanılır. Üçlü arı algoritması, bal arılarının yiyecek arama davranışından esinlenen bir algoritmadır. [60]'ta, tekrarlayan bir sinir ağını optimize etmek için ACO kullanılır. [56], [61], [62] ve [63]'te, yinelemeli sinir ağlarını optimize etmek için sırasıyla orijinal Çiçek Tozlaşma Algoritması (FPA), gri kurt optimizasyoncusu, balina optimizasyon algoritması ve yapay arı kolonisi kullanılır. Çiçek tozlaşma algoritması, algoritmanın keşif ve sömürü aşamaları olarak kullanılan çiçeklerin çapraz tozlaşma ve kendi kendine tozlaşma süreçlerine dayanır. [57]'de, guguk kuşu arama algoritmasının bir hibriti kullanılır. Guguk kuşu arama algoritması, guguk kuşlarının parazitik davranışından esinlenmiştir.

Tablo 2. Yinelemeli sinir Ağlarını Eğitiminde Kullanılan Algoritmaların Karşılaştırması.

Makale	Algoritma	Popülasyon büyüklüğü	Yineleme sayısı	YSA'nın RMSE Aralığı
[55]	Orijinal PSO	30		96.7–97.5 (doğruluk)
[61]	Orijinal PSO, FPA	25-40	50-500	83–86, 84–88 (doğruluk)
[54]	Hibrit PSO	30	100	2.170E-09–0.04243
[69]	BAPSO	45812	100000	0.05–0.1253
[59]	Modified PSO			102.90–441.94
[64]	Hibrit PSO	30	300	0.0237–0.0359
[65]	PSO-CS		1000	1.48E-03–0.0173
[53]	Orijinal CFSO	45840	300	0.05412–2.31560
[58]	Enhanced Ternary Bees	5	100	81.0–99.0 (doğruluk)
[60]	Orijinal ACO	5, 10, 25	100	94.90 (doğruluk)
[62]	Orijinal GWO	8, 15	20	145–162 (belirsizlik)
[56]	Orijinal WOA			0.31–0.32
[63]	Orijinal ABC			25.95–1.64E02
[57]	Hibrit Cuckoo Search		1000	0.0224–0.197

Tablo 2, kullanılan popülasyon büyüklüğü ve maksimum yineleme sayısı ve eğitilen sinir ağının nihai RMSE'si açısından yinelemeli sinir ağlarını eğitmek için kullanılan farklı sürü zekası algoritmalarının bir karşılaştırmasını göstermektedir.

Tablo 2'den, yinelemeli sinir ağlarının orijinal PSO'dan daha fazla PSO hibritleri kullanılarak optimize edildiği çıkarımı yapılabilir. Sadece iki RNN orijinal PSO kullanılarak optimize edilmişken, beş RNN beş farklı PSO hibriti kullanılarak eğitilmiştir. Ancak, kullanılan diğer sürü zekası algoritmaları için, çoğunlukla orijinal olanlar RNN'leri optimize etmek için kullanılmıştır.

FNN'lere benzer şekilde, RNN'leri optimize etmek için sürü zekası algoritmaları tarafından farklı popülasyon büyüklükleri ve yineleme sayıları kullanılır. Ayrıca, en çok kullanılan performans metriği RNN'nin RMSE'sidir, bunu RNN'nin doğruluğu takip eder.

3.3. Evrişimsel Sinir Ağları

[66] ve [67]'de, orijinal PSO evrişimsel sinir ağlarını optimize etmek için kullanılır. [68]'de, PSO'daki ortogonal köşegenleştirme sürecinin kullanımını içeren Ortogonal Öğrenme PSO (OLPSO) adı verilen bir PSO hibriti kullanılır. [70]'te, azalan bir zaman fonksiyonu atalet ağırlığını içeren Geliştirilmiş PSO (IPSO) adı verilen bir PSO hibriti kullanılır. [72]'de, küresel en iyi parçacığın kullanımını hariç tutan ve her parçacık için bir arkadaş listesi içeren değiştirilmiş bir PSO, bir CNN'i optimize etmek için kullanılır.

[71]'de, orijinal ACO bir CNN'i optimize etmek için kullanılır. [67]'de, orijinal PSO'ya ek olarak, orijinal ABC, Bakteriyel Yiyecek Arama Optimizasyonu (BFO), ateş böceği algoritması, Havai Fişek Optimizasyon Algoritması (FOA), Harmoni Arama Algoritması (HSA) ve Yerçekimsel Arama Algoritması (GSA), CNN'leri optimize etmek için kullanılmıştır. BFO, kimyasal gradyanları tespit ederek besinlere yaklaşan belirli bakterilerin davranışına dayanır. FOA, havai fişeklerin patlama şekline dayanır. HSA, caz müzisyenlerinin bir şarkının doğru harmonisini keşfetmek için sesi nasıl uyarladığına dayanır. GSA, yerçekimi ve kütle çekim yasalarından esinlenmiştir.

Tablo 3. Evrişimsel Sinir Ağlarını Eğitiminde Kullanılan Algoritmaların Karşılaştırması.

Makale	Algoritma	Popülasyon büyüklüğü	Yineleme sayısı	YSA'nın RMSE Aralığı
[66]	Orijinal PSO	6	5	71.15–95.16 (doğruluk)
[67]	Orijinal PSO, ABC, BFO, Firefly, FOA, HSA, GSA	300	300	92.27–93.71, 91.63–93.02, 90.96–92.51, 89.10–91.56, 90.27–91.06, 92.22–93.11, 92.34–93.43 (doğruluk)
[68]	OLPSO	5	6	97.7–98.2 (doğruluk)
[70]	IPSO	50	1000	98.85 (doğruluk)
[72]	Modified PSO			0.18–16.38 (hata oranı)
[71]	Orijinal ACO	42370	50	0.39–12.70 (hata oranı)

Tablo 3, kullanılan popülasyon büyüklüğü ve maksimum yineleme sayısı ve eğitilen sinir ağının nihai RMSE'si açısından evrişimsel sinir ağlarını eğitmek için kullanılan farklı sürü zekası algoritmalarının bir karşılaştırmasını göstermektedir.

Tablo 3'ten, üç çalışmanın CNN'leri optimize etmek için sürü zekası algoritmalarının orijinal versiyonlarını kullandığı ve üç çalışmanın hibrit algoritmalar kullandığı görülebilir. CNN'in doğruluğu, performansını değerlendirmek için en yaygın kullanılanıdır, bunu CNN'in hata oranı takip eder.

4. Denetimli Öğrenme Görevleri İçin Sinir Ağları Optimizasyonu

Yapay sinir ağları, denetimli, denetimsiz veya pekiştirmeli öğrenme görevleri için kullanılabilir. Denetimli öğrenmede, sinir ağını eğitmek için girdilerden ve doğru çıktılardan oluşan etiketli veri setleri kullanılır. Tersine, denetimsiz öğrenmede, verilen verilerden örüntüleri keşfetmeye çalışan sinir ağlarını eğitmek için etiketsiz veri setleri kullanılır. Pekiştirmeli öğrenmede, sinir ağı, her bir girdi verisi için birkaç çıktıyı göz önünde bulundurarak en yüksek kümülatif ödüle yol açan bir eylem dizisi alacak şekilde eğitilir.

Ancak, sinir ağlarını optimize etmek için sürü zekası algoritmalarını kullanan çalışmaların, YSA'ları yalnızca denetimli öğrenme için kullandığını bulduk. Sınıflandırma ve regresyon olmak üzere iki tür denetimli öğrenme görevi vardır. Sınıflandırmada çıktı, tahmin edilen ayrık etiketli bir sınıfken, regresyonda çıktı tahmin edilen sürekli bir değerdir.

[20], [21], [24], [25], [27], [37], [39], [41], [43], [46], [47], [48], [49] ve [50]'de sürü zekası algoritmaları tarafından eğitilen ileri beslemeli sinir ağları sınıflandırma problemlerine uygulanmıştır. [20]'de, YSA, UCI makine öğrenmesi deposundan bazı kıyaslama sınıflandırma problemleri, yani 3-bit XOR, iris, balon, meme kanseri, cam ve kalp sınıflandırma problemleri için kullanılmıştır. [21], [24] ve [25]'te, YSA'lar sırasıyla MRI beyin görüntüsü sınıflandırması, bilgisayarlı tomografi görüntülerinden akciğer lezyonlarının tespiti ve sonar hedeflerinin sınıflandırması için kullanılmıştır. [27] ve [47]'de, YSA'lar kıyaslama sınıflandırma problemlerine uygulanmıştır; [27]'de XOR, balon ve kalp kıyaslama sınıflandırma problemleri kullanılırken, [47]'de 3-bit parite, XOR6, XOR9, XOR13 ve 4-bit kodlayıcı/kod çözücü sınıflandırma problemleri kullanılmıştır. [37]'de, YSA, yunusların türünü ıslık sinyallerine göre sınıflandırmak için kullanılır. [48]'de, YSA bir bölgenin sel yarıntısı erozyonu duyarlılığı için kullanılır. [39], [41] ve [49]'da, YSA'lar sırasıyla bir saldırı tespit sistemindeki aktivitelerin sınıflandırılması, oltalama web sitelerinin tespiti ve meyvelerin sınıflandırılması için kullanılır. [43]'te, YSA kıyaslama sınıflandırma problemlerine, yani balon, 3-bit XOR, hepatit, iris, meme kanseri, şarap, monk, kalp, kan ve Avustralya sınıflandırma problemlerine uygulanır. [50]'de, YSA kablosuz iletişim sistemlerinde dijital modülasyon tanıma için kullanılır. [46]'da, YSA, UCI makine öğrenmesi deposundan cam, iris, balon ve meme kanseri kıyaslama sınıflandırma problemlerine uygulanır.

[22], [23], [26], [28], [29], [30], [31], [32], [33], [34], [35], [36], [38], [40], [42], [44], [45], [51] ve [52]'de, sürü zekası algoritmaları tarafından eğitilen ileri beslemeli sinir ağları regresyon problemleri için kullanılmıştır. [22] ve [23]'te, YSA'lar sırasıyla nesne yönelimli sistemlerde hata tahmini ve normal ve yüksek dayanımlı betona gömülü kanal konektörlerinin davranış tahmini için kullanılır. [26], [28] ve [29]'da, YSA'lar sırasıyla iki katmanlı temel zeminleri üzerindeki temellerin taşıma kapasitesini, konut binalarının ısıtma yükünü ve bir konut inşaat projesindeki zemin sıkışma katsayısını tahmin etmek için kullanılır. [30], [36] ve [40]'ta, YSA'lar borsa tahmini için kullanılır. [31], [32], [33] ve [51]'de, YSA'lar sırasıyla bir bölgenin heyelan duyarlılığını, yeraltı su kaynaklarındaki ağır metal konsantrasyonunu, araçların yay yorulma ömrünü ve demiryolu raylarının alt temel katmanları için zeminlerin taşıma oranını tahmin etmek için kullanılır. [34]'te, hidrolojik tahmin için ve [52]'de güneş enerjisi üretimi tahmini için kullanılır. [35]'te, tandem dalgakıranın iç geleneksel moloz yığın dalgakıranının hasar seviyesini tahmin etmek için kullanılır. [38]'de, YSA, UCI makine öğrenmesi deposundan aşağıdaki kıyaslama regresyon problemlerine uygulanır: Kanat Profili Kendi Kendine Gürültü, Beton Çökme Testi (Slump), İstanbul Menkul Kıymetler Borsası, Bilgisayar Donanımı, Otomatik MPG, Orman Yangınları, Meme Kanseri Wisconsin (Prognostik), Kombine Çevrim Güç Santrali, Enerji verimliliği (Isıtma Yükü), Challenger ABD Uzay Mekiği O-Halkası, Doğurganlık, Beton Basınç Dayanımı, Beton Çökme Testi (Dayanım), Beton Çökme Testi (Akış) ve Enerji verimliliği (Soğutma Yükü). [713]'te, YSA güneş ışınımındaki değişimi tahmin etmek için kullanılır. [44]'te, geri dönüştürülmüş agrega betonu için karbonatlaşma derinliğini tahmin etmek için ve [45]'te zaman serisi tahmini için kullanılır.

[55], [57], [58] ve [65]'te, sürü zekası algoritmaları tarafından eğitilen yinelemeli sinir ağları sınıflandırma problemleri için kullanılmıştır. [58]'de, RNN duygu sınıflandırması için kullanılır ve [55]'te voltaj kararsızlığı tahmini için kullanılır. [65]'te, RNN, UCI makine öğrenmesi deposundan iris, cam ve diyabet kıyaslama sınıflandırma problemleri için kullanılır ve [57]'de, iris, Wisconsin meme kanseri, tiroid, diyabet, cam, Avustralya kredi kartı onayı ve 7-bit parite kıyaslama sınıflandırma problemleri için kullanılır.

[53], [54], [56], [59], [60], [61], [62], [63], [64] ve [69]'da, sürü zekası algoritmaları tarafından eğitilen yinelemeli sinir ağları regresyon problemleri için kullanılmıştır. [53]'te, RNN, çift tank modeli adı verilen bir kıyaslama regresyon problemi için kullanılır. [54]'te, RNN güç elektroniği sistemlerinde empedans tanımlaması için kullanılır. [69]'da, gen düzenleyici ağların zamansal genetik ifadeden tersine mühendisliği için kullanılır. [56], [59], [60] ve [61]'de, RNN'ler sırasıyla zaman serisi tahmini, hava kirliliği tahmini, borsa tahmini ve tarımsal ürünlerin verim tahmini için kullanılır. [62]'de, Penn Tree Bank veri setini kullanarak dil modelleme için kullanılır. [63] ve [64]'te,

RNN'ler sırasıyla trafik hacmi tahmini ve kayma dalgası hızı tahmini için kullanılır.

[66], [67], [68], [70], [71] ve [72]'de, sürü zekası algoritmaları tarafından eğitilen evrişimsel sinir ağları sınıflandırma problemlerine uygulanmıştır. Bir regresyon problemi için kullanılan, sürü zekası algoritması tarafından eğitilmiş hiçbir CNN yoktur. [66] ve [71]'de, CNN'ler sınıflandırma veri setlerine uygulanır. [66]'da, CIFAR-10, CIFAR100 ve ILSVRC-2012 veri setleri kullanılır ve [71]'de MNIST, Fashion-MNIST ve CIFAR-10 veri setleri kullanılır. [67]'de, CNN, bilgisayarlı tomografi taramalarında pulmoner kanserli nodüllerin tespitinde kullanılır ve [68]'de dijital görüntüler kullanılarak bitki hastalığı sınıflandırması için kullanılır. [70]'te, beyin MRI görüntülerinde multipl skleroz lezyon segmentasyonuna uygulanır. [72]'de, CNN görüntü sınıflandırması için kullanılır.

Tablo 4. Sürü Zekası Algoritmaları tarafından eğitilen YSA'ların Uygulamalarının Karşılaştırması.

Makale	Uygulama	Kullanılan YSA	YSA RMSE
[20]	3bit -XOR, iris, balon, meme kanseri, cam, kalp kıyaslama sınıflandırma problemleri	MLP	1.292E-22–2.356E-02
[21]	MRI beyin görüntüsü sınıflandırması	MLP	85.0 (doğruluk)
[24]	Akciğer lezyonu tespiti	MLP	97.2–99.5 (doğruluk)
[25]	Sonar hedefleri sınıflandırması	MLP	92.2–96.7 (doğruluk)
[27]	XOR, balon, kalp kıyaslama sınıflandırma problemleri	MLP	2.341E-16–0.3382
[47]	XOR6, XOR9, XOR13, 3-bit parite ve 4-bit kodlayıcı/kod çözücü sınıflandırma problemleri	Deep FNN	4.0130E-06–0.002317
[37]	Yunus ıslıkları sınıflandırması	MLP	0.01414–0.5226
[48]	Sel yarıntısı erozyonu duyarlılığı	Deep FNN	0.89 (AUC)
[39]	Saldırı tespit sistemi aktiviteleri sınıflandırması	MLP	98.0 (doğruluk)
[49]	Oltalama web siteleri tespiti	Deep FNN	96.5 (doğruluk)
[41]	Meyve sınıflandırması	MLP	0.1584

[43]	Kıyaslama sınıflandırma problemleri	MLP	2.381E-10–0.5617
[50]	Dijital modülasyon tanıma	Deep FNN	95.0 (doğruluk)
[46]	İris, balon, cam, meme kanseri kıyaslama sınıflandırma problemi	MLP	0.0991–0.78259
[22]	Nesne yönelimli sistemlerde hata tahmini	MLP	20.7–96.2 (doğruluk)
[23]	Kanal konektörleri davranış tahmini	MLP	14.4–2.49
[26]	Temel taşıma kapasitesi tahmini	MLP	0.3422, 0.3674
[28]	Isıtma yükü tahmini	MLP	2.2899–2.4459
[29]	Zemin sıkışma katsayısı tahmini	MLP	0.0267
[30]	Borsa tahmini	MLP	1.195E-05–0.225 (Ortalama Mutlak Hata)
[31]	Heyelan duyarlılığı tahmini	MLP	0.4444
[32]	Ağır metal konsantrasyonu tahmini	MLP	0.082–0.273
[33]	Yay yorulma ömrü tahmini	MLP	0.6391
[51]	Zemin taşıma oranı tahmini	ELM	0.6391
[34]	Hidrolojik tahmin	MLP	635.5932–1501.1942
[35]	Moloz yığını dalgakıran hasar seviyesi tahmini	MLP	0.515–1.264
[36]	Borsa tahmini	MLP	1.59E-02–7.78E-02
[52]	Güneş enerjisi üretimi tahmini	RBF	5.693E-03–8.149E-03
[38]	Kıyaslama regresyon problemi	MLP	0.09798–0.4472
[40]	Borsa tahmini	MLP	0.03162
[42]	Güneş ışınımı tahmini	MLP	0.1911–0.2032
[44]	Karbonatlaşma derinliği tahmini	MLP	2.281–2.456
[45]	Zaman serisi tahmini	MLP	1.0–3.0 (ortalama hata)
[58]	Duygu sınıflandırması	Deep RNN	81.0–99.0 (doğruluk)
[55]	Voltaj kararsızlığı tahmini	Elman RNN	96.7–97.5 (doğruluk)
[65]	İris, cam, diyabet kıyaslama sınıflandırma problemi	Jordan RNN	1.48E-03–0.0173
[57]	Kıyaslama sınıflandırma problemi	Elman RNN	0.0224–0.197

[53]	Çift tank modeli kıyaslama regresyon problemi	NARX	0.05412–2.31560
[54]	Empedans tanımlaması	Elman RNN	2.170E-09–0.04243
[69]	Gen düzenleyici ağlar tersine mühendisliği	RNN	0.05–0.1253
[59]	Zaman serisi tahmini	Deep RNN	102.90–441.94
[60]	Hava kirliliği tahmini	LSTM	94.90 (doğruluk)
[61]	Borsa tahmini	LSTM	83–88 (doğruluk)
[62]	Dil modelleme	LSTM	145–162 (belirsizlik)
[56]	Tarımsal ürün verim tahmini	Elman RNN	0.31–0.32
[63]	Trafik hacmi tahmini	LSTM	25.95–1.64E02
[64]	Kayma dalgası hızı tahmini	LSTM	0.0237–0.0359
[66]	CIFAR-10, CIFAR100, ILSVRC-2012 sınıflandırma veri seti	VGGNet, ResNet, GoogLeNet	71.15–95.16 (doğruluk)
[71]	MNIST, Fashion-MNIST, CIFAR-10 sınıflandırma veri seti	CNN	0.39–12.70 (hata oranı)
[67]	Pulmoner kanserli nodül tespiti	U-Net	89.10–93.43 (doğruluk)
[68]	Bitki hastalığı sınıflandırması	VGGNet	97.7–98.2 (doğruluk)
[70]	Multipl skleroz lezyon segmentasyonu	CNN	98.85 (doğruluk)
[72]	Görüntü sınıflandırması	CNN	0.18–16.38 (hata oranı)

Tablo 4, sürü zekası algoritmaları tarafından optimize edilen YSA'ların, YSA'nın uygulaması, kullanılan YSA türü ve YSA'nın performansı, yani nihai RMSE'si açısından bir karşılaştırmasını sunar. RMSE'nin sinir ağının performansını değerlendirmek için kullanılmadığı durumlarda, kullanılan performans metriği parantez içinde belirtilmiştir.

Tablo 4'ten, sürü zekası algoritmaları tarafından optimize edilen ileri beslemeli sinir ağlarının, özellikle MLP'lerin, farklı sınıflandırma ve regresyon problemleri için yaygın olarak kullanıldığı görülebilir. Sürü zekası algoritmaları tarafından optimize edilen RNN'ler, ağırlıklı olarak regresyon problemleri ve birkaç sınıflandırma problemi için kullanılmıştır. Ancak, sürü zekası algoritmaları tarafından optimize edilen CNN'ler yalnızca sınıflandırma problemleri için kullanılmış ve herhangi bir regresyon problemi için kullanılmamıştır.

5. Sonuç

Yapay sinir ağları, çeşitli uygulamalar için faydalı olduklarından çok sayıda alanda yaygınlaşmaktadır. Bir YSA kullanmadan önce, YSA'nın türü ve temel olarak katman sayısı ve her katmandaki nöron sayısından oluşan mimarisi gibi belirli karakteristikleri belirlemek önemlidir. Ayrıca, ağın optimal olanlarını bulmak için YSA'nın bağlantı ağırlıkları ve biaslarının ayarlandığı yerde YSA'nın eğitilmesi gerekir. Kullanılacak ana YSA türü genellikle uygulamanın doğasına göre seçilir. Ancak, mimari genellikle deneme-yanılma temelinde belirlenir ve eğitim gradyan iniş algoritmaları kullanılarak yapılır.

Son zamanlarda, sürü zekası algoritmalarının, sinir ağlarının mimarisini belirlemek veya sinir ağlarının eğitimi için yararlı ve avantajlı olduğu bulunmuştur. Bunun nedeni, ağın mimarisini deneme-yanılma yoluyla belirlemek yerine daha akıllı bir yöntemin kullanılmasıdır. Ayrıca, sinir ağlarının eğitimi için, geleneksel gradyan iniş algoritmaları genellikle yerel arama algoritmalarıdır ve ağ için en optimal bağlantı ağırlıklarını ve biaslarını bulmakta yetersizdirler. Dolayısıyla, sürü zekası algoritmaları, hem keşif hem de sömürü yetenekleri sayesinde sinir ağlarının optimal ağırlıklarını ve biaslarını belirlemek için daha iyi seçeneklerdir. Çeşitli alanlarda çeşitli uygulamalar için farklı sinir ağlarını optimize etmek amacıyla bir dizi sürü zekası algoritması kullanılmıştır. Ancak, bildiğimiz kadarıyla, sürü zekası algoritmaları kullanılarak optimize edilen yapay sinir ağları üzerine kapsamlı bir inceleme bulunmamaktadır.

Bu makalede, sürü zekası algoritmaları tarafından optimize edilen farklı YSA türlerinin, YSA'ların optimize edilme şeklinin, ağın optimizasyonu için kullanılan sürü zekası algoritmalarının ve sürü zekası algoritmaları tarafından eğitilen YSA'ların uygulamalarının bir incelemesini sunuyoruz.

Üç ana YSA türü arasında, ileri beslemeli sinir ağlarının sürü zekası algoritmaları kullanılarak en çok optimize edildiği, evrişimsel sinir ağlarının ise sürü zekası algoritmaları kullanılarak en az optimize edildiği bulunmuştur. İleri beslemeli sinir ağları arasında, çok katmanlı algılayıcılar çoğunlukla sürü zekası algoritmaları aracılığıyla optimize edilmiştir. Ayrıca, FNN'ler ve RNN'ler için sürü zekası algoritmalarının eğitim sırasında ağırlıklarını ve biaslarını optimize etmek için en yaygın şekilde kullanıldığı görülebilir. Ancak, CNN'ler için sürü zekası algoritmaları çoğunlukla ağın optimal yapısını seçmek için kullanılmıştır. Farklı sürü zekası algoritmaları arasında, parçacık sürü optimizasyonu ve hibritleri çoğunlukla YSA'ları optimize etmek için kullanılmıştır. Uygulamalar açısından, optimize edilen YSA'lar sadece denetimli öğrenme görevleri, yani sınıflandırma veya regresyon için kullanılmıştır. Optimize edilen FNN'ler ve RNN'ler farklı sınıflandırma ve

regresyon uygulamaları için kullanılmıştır. Ancak, optimize edilen CNN'ler sadece sınıflandırma uygulamaları için kullanılmıştır.

Gelecek çalışmalar için, farklı sürü zekası algoritmaları kullanılarak optimize edilen YSA'ların performansı, YSA'ların aynı uygulama için kullanılmasıyla karşılaştırılabilir, farklı türdeki sinir ağlarını optimize etmek için daha yeni ve yüksek performanslı sürü zekası algoritmaları kullanılabilir ve optimize edilen YSA'lar diğer uygulamalar için kullanılabilir.

Referanslar

[1] M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of Machine Learning. Cambridge, MA, USA: MIT Press, 2018.
[2] N. Shahid, T. Rappon, and W. Berta, "Applications of artificial neural net- works in health care organizational decision-making: A scoping review," PLOS ONE, vol. 14, no. 2, Feb. 2019, Art. no. e0212356.
[3] J. Runge and R. Zmeureanu, "Forecasting energy use in buildings using artificial neural networks: A review," Energies, vol. 12, no. 17, p. 3254, Aug. 2019.
[4] A. Escamilla-García, G. M. Soto-Zarazúa, M. Toledano-Ayala, E. Rivas-Araiza, and A. Gastélum-Barrios, "Applications of artificial neural networks in greenhouse technology and overview for smart agriculture development," Appl. Sci., vol. 10, no. 11, p. 3835, 2020.
[5] B. Li, C. Delpha, D. Diallo, and A. Migan-Dubois, "Application of artificial neural networks to photovoltaic fault detection and diagno- sis: A review," Renew. Sustain. Energy Rev., vol. 138, Mar. 2021. Art. no. 110512.
[6] A. R. Pazikadin, D. Rifai, K. Ali, M. Z. Malik, A. N. Abdalla, and M. A. Faraj, "Solar irradiance measurement instrumentation and power solar generation forecasting based on artificial neural networks (ANN): A review of five years research trend," Sci. Total Environ., vol. 715, May 2020, Art. no. 136848.
[7] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedfor- ward neural networks: A review of two decades of research," Eng. Appl. Artif. Intell., vol. 60, pp. 97-116, Apr. 2017.
[8] M. J. Reddy and D. N. Kumar, "Evolutionary algorithms, swarm intelli- gence methods, and their applications in water resources engineering: A state-of-the-art review," H2Open J., vol. 3, no. 1, pp. 135-188, Jan. 2020.
[9] Q.-V. Pham, D. C. Nguyen, S. Mirjalili, D. T. Hoang, D. N. Nguyen, P. N. Pathirana, and W.-J. Hwang, "Swarm intelligence for next- generation wireless networks: Recent advances and applications," 2020, arXiv:2007.15221.
[10] H. R. Boveiri, R. Khayami, M. Elhoseny, and M. Gunasekaran, "An effi- cient swarm-intelligence approach for task scheduling in cloud-based Internet of Things applications," J. Ambient Intell. Hum. Comput., vol. 10, no. 9, pp. 3469-3479, 2019.
[11] B. A. S. Emambocus, M. B. Jasser, and A. Amphawan, "Towards an optimized channel estimation in optical spatial multiplexing systems via swarm intelligence algorithms," in Proc. IEEE 13th Control Syst. Graduate Res. Colloq. (ICSGRC), Jul. 2022, pp. 77-82.
[12] L. Peška, T. M. Tashu, and T. Horváth, "Swarm intelligence techniques in recommender systems-A review of recent research," Swarm Evol. Comput., vol. 48, pp. 201-219, Aug. 2019.
[13] S. Mishra, R. Sagban, A. Yakoob, and N. Gandhi, "Swarm intelligence in anomaly detection systems: An overview," Int. J. Comput. Appl., vol. 43, no. 2, pp. 109-118, Feb. 2021.
[14] G. Soni, V. Jain, F. T. S. Chan, B. Niu, and S. Prakash, "Swarm intelli- gence approaches in supply chain management: Potentials, challenges and future research directions," Supply Chain Manag., Int. J., vol. 24, no. 1, pp. 107-123, Jan. 2019.
[15] E. Figueiredo, M. Macedo, H. V. Siqueira, C. J. Santana, A. Gokhale, and C. J. A. Bastos-Filho, "Swarm intelligence for clustering-A systematic review with

new perspectives on data mining." Eng. Appl. Artif. Intell., vol. 82, pp. 313-329, Jun. 2019.
[16] L. Brezočnik, I. Fister, and V. Podgorelec, "Swarm intelligence algorithms for feature selection: A review," Appl. Sci., vol. 8, no. 9, p. 1521, Sep. 2018.
[17] B. A. S. Emambocus, M. B. Jasser, M. Hamzah, A. Mustapha, and A. Amphawan, "An enhanced swap sequence-based particle swarm optimization algorithm to solve TSP," IEEE Access, vol. 9. pp. 164820-164836, 2021.
[18] B. A. S. Emambocus, M. B. Jasser, A. Amphawan, and A. W. Mohamed, "An optimized discrete dragonfly algorithm tackling the low exploita- tion problem for solving TSP" Mathematics, vol. 10, no. 19, p. 3647. Oct. 2022.
[19] B. A. S. Emambocus, M. B. Jasser, A. Mustapha, and A. Amphawan, "Dragonfly algorithm and its hybrids: A survey on performance, objectives and applications," Sensors, vol. 21, no. 22, p. 7542, 2021.
[20] W. A. H. M. Ghanem and A. Jantan, "A cognitively inspired hybridization of artificial bee colony and dragonfly algorithms for training multi-layer perceptrons," Cogn. Comput., vol. 10, no. 6, pp. 1096-1134, Dec. 2018.
[21] A. T. Abdulameer, "An improvement of MRI brain images classification using dragonfly algorithm as trainer of artificial neural network," Ibn AL- Haitham J. Pure Appl. Sci., vol. 31, no. 1, pp. 268-276, May 2018.
[22] N. Kayarvizhy, S. Kanmani, and V. Uthariaraj, "Ann models optimized using swarm intelligence algorithms," WSEAS Trans. Comput., vol. 13, pp. 501-519, Jan. 2014.
[23] M. Shariati, M. S. Mafipour, P. Mehrabi, A. Bahadori, Y. Zandi, M. N. A. Salih, H. Nguyen, J. Dou, X. Song, and S. Poi-Ngian, "Appli- cation of a hybrid artificial neural network-particle swarm optimization (ANN-PSO) model in behavior prediction of channel shear connectors embedded in normal and high-strength concrete," Appl. Sci., vol. 9, no. 24, p. 5534, Dec. 2019.
[24] E. E. Nithila and S. S. Kumar, "Automatic detection of solitary pul- monary nodules using swarm intelligence optimized neural networks on CT images," Eng. Sci. Technol., Int. J., vol. 20, no. 3, pp. 1192-1202, 2017.
[25] M. Khishe and A. Safari, "Classification of sonar targets using an MLP neural network trained by dragonfly algorithm," Wireless Pers. Commun., vol. 108, no. 4, pp. 2241-2260, Oct. 2019.
[26] H. Moayedi, M. M. Abdullahi, H. Nguyen, and A. S. A. Rashid, "Com- parison of dragonfly algorithm and Harris hawks optimization evolution- ary data mining techniques for the assessment of bearing capacity of footings over two-layer foundation soils," Eng. Comput., vol. 37, no. 1, pp. 437-447, Jan. 2021.
[27] J. Xu and F. Yan, "Hybrid Nelder-Mead algorithm and dragonfly algo- rithm for function optimization and the training of a multilayer percep- tron," Arabian J. Sci. Eng., vol. 44, no. 4, pp. 3473-3487, Apr. 2019.
[28] H. Moayedi, H. Nguyen, and L. Foong, "Nonlinear evolutionary swarm intelligence of grasshopper optimization algorithm and gray wolf opti-mization for weight adjustment of neural network," Eng. Comput., vol. 37, pp. 1265-1275, Apr. 2021.
[29] D. Tien Bui, V.-H. Nhu, and N.-D. Hoang, "Prediction of soil com- pression coefficient for urban housing project using novel integration machine learning approach of swarm intelligence and multi-layer per- ceptron neural network," Adv. Eng. Informat., vol. 38, pp. 593-604, Oct. 2018.

[30] R. Jamous, H. ALRahhal, and M. El-Darieby, "A new ANN-particle swarm optimization with center of gravity (ANN-PSOCOG) prediction model for the stock market under the effect of COVID-19," Sci. Program., vol. 2021, pp. 1-17, Apr. 2021.
[31] D. T. Bui, H. Moayedi, B. Kalantar, A. Osouli, B. Pradhan, H. Nguyen, and A. S. A. Rashid, "A novel swarm intelligence-Harris hawks optimization for spatial assessment of landslide susceptibility," Sensors, vol. 19, no. 16, . 3590, 2019.
[32] M. Alizamir and S. Sobhanardakani, "An artificial neural network-particle swarm optimization (ANN- PSO) approach to predict heavy metals contamination in groundwater resources," Jundishapur J. Health Sci., vol. 10, no. 2, Apr. 2018, Art. no. e67544.
[33] Y. S. Kong. S. Abdullah, D. Schramm, M. Z. Omar, and S. M. Haris, "Design of artificial neural network using particle swarm optimisation for automotive spring durability." J. Mech. Sci. Technol., vol. 33, no. 11. pp. 5137-5145, Nov. 2019.
[34] W.-J. Niu, Z.-K. Feng, B.-F. Feng, Y.-S. Xu, and Y.-W. Min, "Parallel computing and swarm intelligence based artificial intelligence model for multi-step-ahead hydrological time series prediction," Sustain. Cities Soc.. vol. 66, Mar. 2021, Art. no. 102686.
[35] G. Kuntoji, S. M. Rao, and E. N. B. Reddy, "Prediction of damage level of inner conventional rubble mound breakwater of tandem breakwater using swarm intelligence-based neural network (PSO-ANN) approach," in Soft Computing for Problem Solving. Springer, 2019, pp. 441-453.
[36] G. Kumar, U. P. Singh, and S. Jain, "Swarm intelligence based hybrid neural network approach for stock price forecasting." Comput. Econ., vol. 60, no. 3, pp. 991-1039, Oct. 2022.
[37] J. Wu, M. Khishe, M. Mohammadi, S. H. T. Karim, and M. Shams, "Acoustic detection and recognition of dolphins using swarm intelligence neural networks," Appl. Ocean Res., vol. 115, Oct. 2021, Art. no. 102837.
[38] Y. Wang, H. Liu, Z. Yu, and L. Tu, "An improved artificial neural net- work based on human-behaviour particle swarm optimization and cellular automata," Expert Syst. Appl., vol. 140, Feb. 2020, Art. no. 112862.
[39] S. Norwahidayah, A. A. Noraniah, N. Farahah, A. Amirah, N. Liyana, and N. Suhana, "Performances of artificial neural network (ANN) and particle swarm optimization (PSO) using KDD cup'99 dataset in intrusion detection system (IDS)," J. Phys., Conf. Ser., vol. 1874, no. 1, May 2021. Art. no. 012061.
[40] F. Ghashami, K. Kamyar, and S. A. Riazi, "Prediction of stock market index using a hybrid technique of artificial neural networks and particle swarm optimization," Appl. Econ. Finance, vol. 8, no. 3, p. 1, Apr. 2021.
[41] Y. Zhang P. Phillips, S. Wang, G. Ji, J. Yang, and J. Wu, "Fruit clas- sification by biogeography-based optimization and feedforward neural network," Expert Syst., vol. 33, no. 3, pp. 239-253, 2016.
[42] A. Aljanad, N. M. L. Tan, V. G. Agelidis, and H. Shareef, "Neural network approach for global solar irradiance prediction at extremely short-time-intervals using particle swarm optimization algorithm," Energies, vol. 14, no. 4, p. 1213, Feb. 2021.

[43] D. Bairathi and D. Gopalani, "Numerical optimization and feed-forward neural networks training using an improved optimization algorithm: Multiple leader salp swarm algorithm," Evol. Intell., vol. 14, no. 3, pp. 1233-1249, Sep. 2021.
[44] K. Liu, M. S. Alam, J. Zhu, J. Zheng, and L.. Chi, "Prediction of car- bonation depth for recycled aggregate concrete using ANN hybridized with swarm intelligence algorithms," Construct. Building Mater., vol. 301. Sep. 2021, Art. no. 124382.
[45] S. A. Abdulkarim and A. P. Engelbrecht, "Time series forecasting with feedforward neural networks trained using particle swarm optimiz- ers for dynamic environments." Neural Comput. Appl., vol. 33, по. 7. pp. 2667-2683, Apr. 2021,
[46] B. A. S. Emambocus. M. B. Jasser, and A. Amphawan, "An optimized continuous dragonfly algorithm using Hill climbing local search to tackle the low exploitation problem," IEEE Access, vol. 10, pp. 95030-95045, 2022.
[47] F. Xu, C.-M. Pun, H. Li, Y. Zhang, Y. Song, and H. Gao, "Training feed-forward artificial neural networks with a modified artificial bee colony algorithm," Neurocomputing, vol. 416, pp. 69-84, Nov. 2020.
[48] S. S. Band. S. Janizadeh, S. Chandra Pal, A. Saha, R. Chakrabortty. M. Shokri, and A. Mosavi, "Novel ensemble approach of deep learning neural network (DLNN) model and particle swarm optimization (PSO) algorithm for prediction of gully erosion susceptibility," Sensors, vol. 20. no. 19, p. 5609, Sep. 2020.
[49] G. Vrbančič, 1. Fister, and V. Podgorelec, "Swarm intelligence approaches for parameter setting of deep learning neural network: Case study on phish- ing websites classification," in Proc. 8th Int. Conf. Web Intell., Mining Semantics, Jun. 2018, pp. 1-8.
[50] W. Shì, D. Liu, X. Cheng, Y. Li, and Y. Zhao, "Particle swarm optimization-based deep neural network for digital modulation recogni- tion," IEEE Access, vol. 7, pp. 104591-104600, 2019.
[51] A. Bardhan, P. Samui, K. Ghosh, A. H. Gandomi, and S. Bhattacharyya, "ELM-based adaptive neuro swarm intelligence techniques for predicting the California bearing ratio of soils in soaked conditions," Appl. Soft Comput., vol. 110, Oct. 2021, Art. no. 107595.
[52] Z. Yang, M. Mourshed, K. Liu, X. Xu, and S. Feng, "A novel compet- itive swarm optimized RBF neural network model for short-term solar power generation forecasting," Neurocomputing, vol. 397, pp. 415-421, Jul. 2020.
[53] G. M. Lozito and A. Salvini, "Swarm intelligence based approach for efficient training of regressive neural networks." Neural Comput. Appl., vol. 32, no. 14, pp. 10693-10704, Jul. 2020.
[54] P. Xiao, G. K. Venayagamoorthy, and K. A. Corzine, "Combined train- ing of recurrent neural networks with particle swarm optimization and backpropagation algorithms for impedance identification," in Proc. IEEE Swarm Intell. Symp., Apr. 2007, pp. 9-15.
[55] A. M. Ibrahim and N. H. El-Amary, "Particle swarm optimization trained recurrent neural network for voltage instability prediction," J. Electr. Syst. Inf. Technol., vol. 5, no. 2, pp. 216-228, 2018.
[56] P. Murali, R. Revathy, S. Balamurali, and A. S. Tayade, "Integration of RNN with GARCH refined by whale optimization algorithm for yield forecasting:

A hybrid machine learning approach," J. Ambient Intell. Hum. Comput., vol. 11, pp. 1-13, Apr. 2020.
[57] N. M. Nawi, A. Khan, M. Z. Rehman, H. Chiroma, and T. Herawan, "Weight optimization in recurrent neural networks with hybrid meta- heuristic cuckoo search techniques for data classification," Math. Prob- lems Eng., vol. 2015, pp. 1-12, Nov. 2015.
[58] S. Zeybek, D. T. Pham, E. Koç, and A. Seçer, "An improved bees algo- rithm for training deep recurrent networks for sentiment classification," Symmetry, vol. 13, no. 8, p. 1347, Jul. 2021.
[59] E. Bas, E. Egrioglu, and E. Kolemen, "Training simple recurrent deep arti- ficial neural network for forecasting using particle swarm optimization," Granular Comput., vol. 7, no. 2, pp. 411-420, Apr. 2022.
[60] G. J. Kuri-Monge, M. A. Aceves-Fernandez, J. A. Ramírez-Montañez, and J. C. Pedraza-Ortega, "Capability of a recurrent deep neural network optimized by swarm intelligence techniques to predict exceedances of airborne pollution (PMx) in largely populated areas," in Proc. Int. Conf. Inf. Technol. (ICIT), Jul. 2021, pp. 61-68.
[61] K. Kumar and T. U. Haider, "Enhanced prediction of intra-day stock market using metaheuristic optimization on RNN-LSTM network," New Gener. Comput., vol. 39, no. 1, pp. 231-272, Apr. 2021.
[62] B. Z. Aufa, S. Suyanto, and A. Arifianto, "Hyperparameter setting of LSTM-based language model using grey wolf optimizer," in Proc. Int. Conf. Data Sci. Its Appl. (ICODSA), Aug. 2020, pp. 1-5.
[63] A. Bosire, "Recurrent neural network training using ABC algorithm for traffic volume prediction," Informatica, vol. 43, no. 4, pp. 551-559, Dec. 2019.
[64] J. Wang, J. Cao, and S. Yuan, "Shear wave velocity prediction based on adaptive particle swarm optimization optimized recurrent neural network," J. Petroleum Sci. Eng., vol. 194, Nov. 2020, Art. no. 107466.
[65] R. Talal and Z. Tariq. "Training recurrent neural networks by a hybrid PSO-cuckoo search algorithm for problems optimization," Int. J. Comput. Appl., vol. 159, no. 3, pp. 32-38, Feb. 2017.
[66] J. Chang, Y. Lu, P. Xue, Y. Xu, and Z. Wei, "Automatic channel pruning via clustering and swarm intelligence optimization for CNN." Appl. Intell.. vol. 52, pp. 17751-17771, Apr. 2022.
[67] C. A. de Pinho Pinheiro, N. Nedjah, and L. de Macedo Mourelle. "Detection and classification of pulmonary nodules using deep learning and swarm intelligence," Multimedia Tools Appl., vol. 79, nos. 21-22, pp. 15437-15465, Jun. 2020.
[68] A. Darwish, D. Ezzat, and A. E. Hassanien, "An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis," Swarm Evol. Com- put., vol. 52, Feb. 2020, Art. no. 100616.
[69] A. Khan, S. Mandal, R. K. Pal, and G. Saha, "Construction of gene regu- latory networks using recurrent neural networks and swarm intelligence," Scientifica, vol. 2016, pp. 1-14, Jan. 2016.
[70] R. Krishna Priya and S. Chacko, "Improved particle swarm optimized deep convolutional neural network with super-pixel clustering for multiple sclerosis lesion segmentation in brain MRI imaging," Int. J. Numer. Methods Biomed. Eng., vol. 37, no. 9, Sep. 2021, Art. no. e3506.

[71] E. Byla and W. Pang, "DeepSwarm: Optimising convolutional neural networks using swarm intelligence," in Proc. UK Workshop Comput. Intell. Cham, Switzerland: Springer, 2019, pp. 119-130.
[72] S. C. Nistor and G. Czibula, "IntelliSwAS: Optimizing deep neural net- work architectures using a particle swarm-based approach," Expert Syst. Appl., vol. 187, Jan. 2022, Art. no. 115945.

Bölüm 7

İŞ KAZALARININ YAPAY SİNİR AĞLARI VE SARIMA MODELİ YAKLAŞIMI İLE TAHMİNİ: TÜRKİYE ÖRNEĞİ

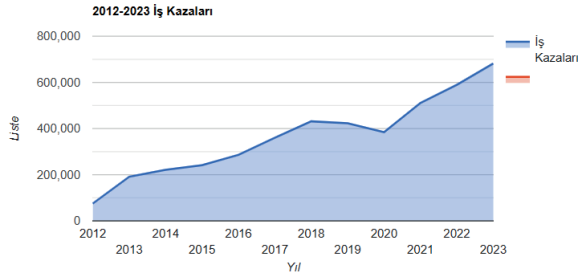
İbrahim Birkan ÖZTÜRK[1]
Serpil TÜRKYILMAZ[2]

1 Lisans Öğrencisi, *Bilecik Şeyh Edebali Üniversitesi*, *Fen Fakültesi, İstatistik ve Bilgisayar Bilimleri Bölümü,* E-mail: birkanozturk_28@hotmail.com, Orcid: https://orcid.org/0009-0003-2321-1062
2 Prof. Dr., *Bilecik Şeyh Edebali Üniversitesi*, *Fen Fakültesi, İstatistik ve Bilgisayar Bilimleri Bölümü,* E-mail: serpil.turkyilmaz@bilecik.edu.tr, Orcid: https://orcid.org/0000-0002-7193-4148

1. GİRİŞ

İş kazaları, çalışanların kendi işyerlerinde veya işleri ile ilgili bir yerde çalışırken uğradığı bedensel zarar veya psikolojik travma sonucu meydana gelen olayları ifade etmektedir. Dünya Sağlık Örgütü (WHO)'nün tanımına göre iş kazaları, "çalışma sırasında meydana gelen ve çalışan üzerinde fiziksel ya da psikolojik zararlar oluşturan, işyerinde ya da işin yapılması esnasında meydana gelen ani olay" şeklinde de tanımlanmıştır. Genellikle ihmallerden, gerekli güvenlik önlemlerinin alınmamasından, insan hatalarından, çalışılan ekipmanların arızalarından veya ağır çalışma koşulları gibi çeşitli sebeplerle meydana gelmektedir. Çalışanlara verilen eğitimlerin yetersizliği iş kazalarına sebebiyetin başlıca faktörlerinden biridir. Çalışanların ekipmanlar hakkında yeterli bilgi sahibi olmaması, güvenlik prosedürleri hakkındaki bilgi eksiklikleri gibi faktörler iş kazalarının başlıca sebeplerindendir. İş yerlerinde çalışanlara, çalışma şartlarına uygun koruyucu ekipman verilmemesi veya çalışma alanında yeterli güvenlik önleminin alınmaması da iş kazalarına neden olabilmektedir. Ayrıca makine, ekipman ve altyapı bakımlarının ihmal edilmesi çalışanların can güvenliğini ciddi oranda tehdit etmektedir. Yanlış kablo tesisatı, kaygan zeminler, arızalı veya kalitesiz ekipmanlar can kaybına neden olabilecek kazaların yaşanma olasılığını arttırmaktadır. Ağır çalışma şartları ve yetersiz dinlenme molaları nedeniyle çalışanların yorgun düşmesi veya stres altında kalması da güvenlik anlamına büyük riskler taşımaktadır. Bu durumlarda bir olaya verilen tepki süresi artabilir, karar verme yetenekleri etkilenebilir ve bu gibi durumlar da iş kazalarına neden olabilmektedir.

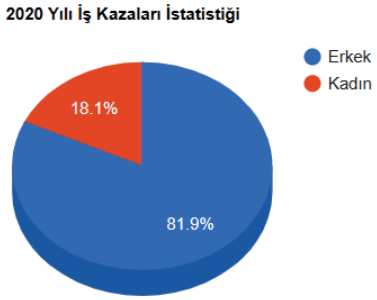
Nedenisg (2021) raporuna göre; Sosyal Güvenlik Kurumu (SGK) tarafından yayınlanan verilere göre Türkiye'de 2020 yılında 4-1/a ve 4-1/b sigortalı çalışan sayısı kapsamında toplam 384.262 iş kazası gerçekleşmiştir. Bu kazalar ve meslek hastalıkları sonucunda 1.245 çalışan hayatını kaybetmiş ve 908 çalışan meslek hastalığına yakalanmıştır. Rapora göre; toplamda 3.492.824 gün süresince geçici iş görememezlik durumu yaşanmıştır.



Şekil 1. *2012-2023 Yılı İş Kazaları Grafiği (https://nedenisguvenligi.com/yillara-gore-is-sagligi-ve-guvenligi-istatistikleri/)*

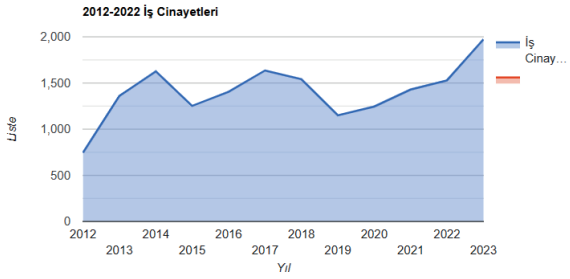
Şekil 1., 2012-2023 yılları arasında yaşanan iş kazalarının zamana bağlı

arttığını özellikle 2020 yılından sonra hızlı bir yükselme gösterdiği gözlenmektedir.



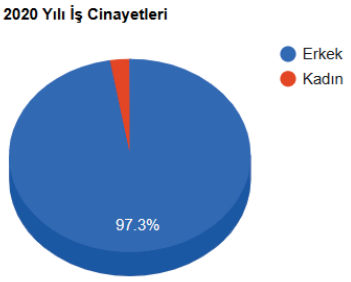
Şekil 2. *2020 Yılı İş Kazalarının Cinsiyete Göre Dağılımı (https://nedenisguvenligi.com/is-kazalari-ve-meslek-hastaliklari-istatistigi-2020/)*

Şekil 2' de 2020 yılında meydana gelen iş kazalarının cinsiyete göre dağılımı iş kazalarının büyük çoğunluğunun (% 81.9) erkek çalışanlar arasında meydana geldiğini göstermektedir.



Şekil 3. *2012-2022 Yılları Arasında Meydana Gelen İş Cinayetleri Grafiği (https://nedenisguvenligi.com/yillara-gore-is-sagligi-ve-guvenligi-istatistikleri/)*

Şekil 3' de 2012-2022 yılları arasında meydana gelen iş cinayetlerinin (iş kazaları sonucu ölümlerin) dağılımını göstermektedir. Grafikten; 2012'den itibaren iş cinayetlerinde genel olarak yıllara göre bir artış eğilimi gözlendiğini söylemek mümkündür.

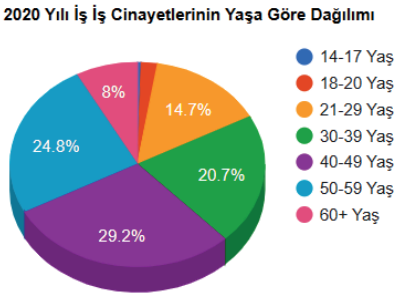


Şekil 4. *2020 Yılı İş Cinayetlerinin Cinsiyete Göre Dağılımı (https://nedenisguvenligi.com/is-kazalari-ve-meslek-hastaliklari-istatistigi-2020/)*

Şekil 4, 2020 yılında iş kazası sonucu meydana gelen ölümlerin cinsiyete göre dağılımını göstermektedir. SGK verilerine göre iş cinayetlerinde 1.245 çalışan hayatını kaybetmiştir. İş kazası sonucu ölümlerin neredeyse tamamının (%97.3) erkek çalışanlar arasında gerçekleştiği görülmektedir.

Şekil 5. *2020 Yılı İş Cinayetlerinin İllere Göre Dağılımı (https://nedenisguvenligi.com/is-kazalari-ve-meslek-hastaliklari-istatistigi-2020/)*

Şekil 5, pasta grafiği 2020 yılında Türkiye'deki iş cinayetlerinin illere göre dağılımını göstermektedir. Grafiğe göre, İstanbul %26.8 ile en yüksek orana sahip, Ankara %14.5 ile ikinci sırada, İzmir %13.4 ile üçüncü sırada yer almaktadır. Bu üç büyükşehir, toplam iş cinayetlerinin yaklaşık %55'ini oluşturmaktadır.



Şekil 6. *2020 Yılı İş Cinayetleri Yaş Dağılımı Pasta Grafiği (https://nedenisguvenligi.com/is-kazalari-ve-meslek-hastaliklari-istatistigi-2020/)*

Şekil 6, 2020 yılı iş kazası sonucu meydana gelen ölümlerin yaş gruplarına göre dağılımını göstermektedir. İş kazası sonucu ölümlerin en çok 40-49 yaş aralığında olduğu görülmektedir. 50-59 yaş grubu ikinci sırada yer almaktadır. En genç (14-17 ve 18-20) ve en yaşlı (60+) çalışanların oranlarının ise daha düşük olduğu görülmektedir. Nedenisg (2021) raporuna göre; 2013-2019 yılları arasında toplamda 2 milyon 537 bin 730 insan iş kazası geçirmiştir. Bu sayı, çalışan nüfusa oranla %6'nın üzerinde bir orana tekabül etmektedir.

1. LİTERATÜR

Dünyada ve Türkiye'de literatürde iş kazalarına ilişkin çok sayıda çalışma mevcuttur. İzleyen kısımda özet olarak bu çalışmalardan bahsedilmektedir. Jacinto ve Aspinwall (2003) çalışmasında, iş kazalarının araştırılması ve raporlanmasına yönelik pratik bir yöntem önermiştir. Geliştirilen WAIT (Work Accidents Investigation Technique) yöntemi, tüm endüstri sektörlerinde aktif hataları, etkileyen faktörleri ve gizli koşulları belirlemeyi ve sınıflandırmayı amaçlamıştır. Yöntem, uzman olmayan kişiler tarafından da uygulanabilir şekilde tasarlanmış ve çeşitli sektörlerde başarıyla test edilmiştir. Yardım vd. (2007) çalışmasında, her yıl insanların engellenebilir iş kazaları ve meslek hastalıkları sonucu yaşamını yitirdiğini veya engelli hale geldiğini vurgulamıştır. 2000-2005 yılları arasında iş kazası ve meslek hastalıklarının ölüm oranlarını incelemiştir. 2005 yılında 73,923 iş kazası olmuş ve 1,096 kişi hayatını kaybetmiştir. Çalışma, iş sağlığı ve güvenliği alanında daha fazla önlem alınması gerektiğini vurgulamıştır ve iş kazalarını azaltmak için kapsamlı politikaların önemi üzerinde durmuştur. Ünal vd., (2008) çalışmasında, iş kazaları ve meslek hastalıklarının ekonomiye ciddi bir yük getirdiğine dikkat çekmiştir. Önlem almanın ne kadar önemli olduğunu ifade etmiştir. Tarım sektöründe bir iş kazasının ortalama maliyeti 7,250$, diğer sektörlerde bu rakam 3,996$ olarak tespit edilmiştir. Genel sektör ortalamasının 4,011$ dolar olduğu sonucuna varılmıştır. Tarım sektöründeki kazaların diğer sektörlere göre %81 daha maliyetli olduğu sonucuna varılmıştır.

Taylan (2008) çalışmasında, Tuzla tersanelerindeki artan iş kazalarını ve ölümleri ele almıştır. Bu kazaların toplumsal rahatsızlığa yol açtığını ve çözümün uzmanlarca yapılan detaylı analizlerle kısa, orta ve uzun vadeli önlemlerin alınması gerektiğini vurgulamıştır. Ayrıca, kazaların nedenleri ve risk faktörleri gözden geçirilmiş, dünyanın farklı yerlerinde yapılan uygulamalar araştırılmış ve bunların Türkiye'ye nasıl uyarlanabileceği önerilmiştir. Ceylan (2011) çalışmasında 2009 yılı SGK verileriyle Türkiye'deki iş kazalarını incelemiş ve Türkiye'nin iş güvenliği durumunu çeşitli gelişmiş ülkelerle karşılaştırmıştır. Karadeniz (2012) çalışmasında, küreselleşme ve sanayileşmenin tehlikelerinin gelişmiş ülkelerden gelişmekte olan ülkelere geçtiğini belirtmiştir. Gelişmekte olan ülkelerde, düşük eğitimli işçilere uyum sağlamanın zor olduğu vurgulanmıştır. Ayrıca, yetersiz iş denetimi nedeniyle iş kazaları ve meslek hastalıklarının arttığı sonucuna varılmıştır. Sosyal koruma eksikliklerinin, gelişmiş ülkelere göre daha fazla olduğu ifade edilmiştir. Ceylan (2014) çalışmasında, Türkiye'de iş kazalarını tahmin etmek için yapay sinir ağları (ANN) kullanmıştır. 1970-2012 verileriyle geliştirilen model, 2025 yılına kadar üç farklı senaryoya göre kaza, ölüm ve kalıcı iş göremezlik tahminleri yapmıştır. En iyi ağ mimarisi olarak 2-5-1 yapısı seçilmiş ve modelin doğruluğu geçmiş verilerle test edilerek uygulanabilirliği doğrulanmıştır. Modelin performansı MAPE, MAE ve RMSE ölçütleriyle değerlendirilmiştir. Mohammadfam vd., (2015) çalışmasında, büyük ölçekli iş yerlerindeki iş kazalarının şiddetini analiz etmek ve tahmin etmek amacıyla yapay sinir ağları (ANN) modeli kullanılmıştır. 2005-2014 yılları arasında 960 iş kazası incelenmiş ve kazaların şiddetini etkileyen faktörler belirlenmiştir. Sonuçlar, ANN'lerin iş kazalarının şiddetini modellemede güvenilir bir yöntem olduğunu göstermiştir. Çavuş ve Taçgın (2016) çalışmasında, Türkiye'deki iş kazalarının özellikle inşaat sektöründe yoğunlaştığını ve bu kazaların sayısal olarak arttığını tespit etmiştir. İnşaat sektöründeki iş kazaları incelenmiş ve oluş biçimlerine göre sınıflandırılmıştır. Ayrıca nedenleri de araştırılmıştır. Çalışma kapsamında, belirli bir dönemde gerçekleşen iş kazaları analiz edilmiş ve inşaat sektöründe iş güvenliğini artırmaya yönelik öneriler sunulmuştur. Gül vd., (2016) çalışmasında, Türkiye'deki farklı sektörlerden 234 iş kazası verisini k-means kümeleme algoritması kullanarak analiz etmiştir. Çalışma, işçi yatırımlarının verimli yönetilmesini ve işçi özellikleri ile kazalar arasındaki ilişkiyi araştırmıştır. Sonuçlar, veri madenciliği tekniklerinin iş sağlığı ve güvenliği alanında kullanılmasının kazaları azaltmada etkili olabileceğini göstermiştir. Bekar vd., (2017) çalışmalarında, Türkiye'de 2005-2014 yılları arasında meydana gelen iş kazalarını ve meslek hastalıklarının ekonomik maliyetlerini incelemiştir. Çalışmada, Çalışma ve Sosyal Güvenlik Bakanlığı'nın verileri kullanılarak iş göremezlik süreleri ve günlük işçi kazançları çarpımıyla maliyet hesabı yapmıştır. Öçal ve Çiçek (2017) çalışmasında, iş kazalarının özellikle 2000'li yıllarda Tuzla, Soma ve Ermenek'teki kazalarla daha fazla gündeme geldiğini belirtmiştir. Türkiye'de iş kazalarının önüne geçmek için yasal dü-

zenlemeler ve önlemler alınmakla birlikte, bu çabaların AB ülkeleriyle kıyaslandığında yetersiz olduğu gözlemlenmiştir. Çalışma, Türkiye'deki iş kazaları ile AB ülkelerindeki durumu kıyaslayarak, iş kazalarına neden olan sorunları ve çözüm önerilerini tartışmıştır. Güğercin ve Baytorun (2018) çalışmasında, iş kazalarının önlenmesi üzerine odaklanmış ve Türkiye'deki iş kazalarının özellikle tarım ve inşaat sektörlerinde yüksek olduğunu vurgulamışlardır. Çalışmalarında, iş sağlığı ve güvenliği sistemlerinin eksikliklerine dikkat çekmiş ve seracılık gibi özel üretim alanlarında daha fazla önlem alınması gerektiğini ifade etmişlerdir. Bayraktaroğlu vd., (2018) çalışmalarında mavi yakalı çalışanların iş sağlığı ve güvenliği (İSG) konusundaki algılarını incelemiştir. Araştırmada Sakarya'daki bir gıda işletmesinde çalışan 120 kişiye anket uygulanmış ve iş güvenliği ile iş kazalarına yönelik algılarının genellikle yüksek olduğu bulunmuştur. Demografik faktörlerden yalnızca yaş, algı düzeyini etkileyen bir faktör olarak öne çıkmıştır. Çalışma, mavi yakalı çalışanlar için iş güvenliği uygulamalarının önemini vurgulamıştır. Şen vd. (2018) çalışmasında, Türkiye ve Avrupa Birliği (AB) ülkelerindeki iş kazalarını karşılaştırmıştır. Veriler SGK ve EUROSTAT'tan alınmış olup, Türkiye'de ölümcül iş kazalarının AB ülkelerine kıyasla çok daha yüksek olduğu tespit edilmiştir. 2014 yılı verilerine göre, Türkiye'deki ölümcül iş kazası oranı AB ortalamasının yaklaşık 6,7 katı olduğu tespit edilmiştir. Çalışma, Türkiye'de özellikle ölümcül iş kazalarını önlemeye yönelik tedbirlerin yetersiz olduğunu vurgulamakta ve bu konuda değerlendirmeler sunmuştur. Ergül (2018) çalışmasında, Türkiye'deki iş kazaları sonucu ölüm ve iş göremezlik verilerini 2016-2020 yılları için Box-Jenkins (ARIMA) Yöntemi ve Yapay Sinir Ağları (YSA) ile modellemiştir. Her iki yöntemin tahmin sonuçları karşılaştırıldığında, YSA ile oluşturulan model daha doğru tahminler sunmuştur. ARIMA' ya göre ölüm ve iş göremezlik sayılarının 2016-2020 dönemi için azalacağı tahmin edilirken, YSA'ya göre bu sayıların artacağı öngörülmüştür. Bilim ve Çelik (2018) çalışmasında, Türkiye'de inşaat sektöründeki iş kazalarının 2012 yılında yürürlüğe sokulan 6331 sayılı İş Sağlığı ve Güvenliği Kanunu'ndan sonra nasıl bir seyir izlediğini araştırmıştır. 2012-2016 yılları arasında tüm sektörlerdeki iş kazaları incelenmiş, inşaat sektörünün bu kazalar içindeki payı ve alt dallardaki durumu analiz edilmiştir. Çalışmada, kaza olabilirlik oranı belirlenerek alt sektörlerle birlikte değerlendirilmiş ve işyeri büyüklüğünün kaza riski üzerindeki etkisi incelenmiştir. Sonuç olarak, inşaat sektöründe kişi başına düşen iş kazası oranının yüksek olduğu tespit edilmiş ve bu oranı azaltmaya yönelik öneriler sunulmuştur.

Ceylan (2021) çalışmasında 2012-2019 yılları arasında Türkiye'nin İş Sağlığı ve Güvenliği (İSG) verilerini inceleyerek iş kazalarına çözüm olarak yeni bir model önerisinde bulunmuştur. Çalış ve Büyükakıncı (2021) çalışmasında, Türkiye'deki iş kazalarını 2013-2016 yılları arasındaki ILOSTAT ve SGK verilerini karşılaştırarak incelemiştir. İş kazalarının sadece çalışanları

değil, aileleri, işyerlerini ve ülkeyi de ciddi şekilde etkilediği vurgulanmış ve iş sağlığı ve güvenliği uygulamalarının yaygınlaştırılması gerektiği belirtilmiştir. Çalışma, Türkiye'deki alınan önlemlerin yetersiz olduğunu ve iş kazalarının arttığını ortaya koymuştur. SGK ve ILOSTAT verileri büyük ölçüde tutarlı bulunmuş, ancak 2016 yılına ait ölümlü olmayan iş kazalarında önemli bir fark gözlemlenmiştir. Bu da Türkiye'de yayımlanan verilerin uluslararası verilerle uyumlu olduğu görülmüştür. Mammadov (2021) çalışmasında, boru hattı inşaatındaki iş kazalarını önlemek için makine öğrenimi algoritmalarını kullanan bir model geliştirmiştir. 1.184 vaka içeren bir veri seti oluşturulmuş ve kazaların nedenleri ile sonuçları analiz edilmiştir. On bir farklı algoritma test edilmiş, en iyi performansı veren Derin Öğrenme yöntemi seçilmiştir. Optimizasyon ile modelin doğruluğu artırılmış ve bu sayede kazaların nedenlerini önceden belirleyerek önleyici tedbirler alınabileceği sonucuna varılmıştır. Akay vd., (2021) çalışmasında, Avrupa Birliği'ne üye olan 23 ülke ve Türkiye'yi iş kazaları açısından benzerlik ve farklılıklarına göre sınıflandırmıştır. 2008-2017 yılları arasında ormancılık ve tomrukçuluk sektörüne ait veriler kullanılarak k-means kümeleme yöntemi uygulanmıştır. Türkiye, Küme 2'de yer almış ve bu kümedeki ülkelerin iş kazası oranlarının diğer kümelere kıyasla daha düşük olduğu belirlenmiştir. Dyreborg vd., (2022) çalışmasında, iş kazalarını öngören değişken olarak kişilik faktörlerini incelemiş ve "Beş Büyük" kişilik modeli kullanarak meta-analiz yapmıştır. Sonuçlar, kişiliğin iş kazaları üzerindeki etkisinin durumsal faktörlere bağlı olarak değişkenlik gösterdiğini ortaya koymuştur. Ancak, düşük uyumluluk düzeyinin iş kazalarına karışma konusunda tutarlı ve genellenebilir bir öngörücü olduğu belirlenmiştir. Özdemir (2024) çalışmasında, Türkiye'de tarım sektörünün stratejik önemini ve istihdama katkısını vurgulamıştır. 2018-2022 yılları arasında sektörde 13.993 iş kazası ve 96 ölümcül kaza meydana gelmiş, en fazla kaza 2022'de yaşanmıştır. Çalışmada, mevcut iş sağlığı ve güvenliği uygulamalarının yetersiz olduğu belirtilmiş ve eğitimlerin yaygınlaştırılması, sağlık taramalarının artırılması ile denetimlerin sıkılaştırılması önerilmiştir. Literatürde iş kazalarının modellenmesi ve tahmininde zaman serileri analizi, yapay sinir ağları, üstel düzeltme metodu, olasılık dağılımları yaklaşımı ve makine öğrenmesi gibi yöntemlerin kullanıldığı çalışmalar mevcuttur. Örneğin, geçmiş yıllara ait iş kazası verileri kullanılarak gelecekteki kaza sayıları tahmin edilebilir. Türkiye'de yapılan bir çalışmada, 1970-2016 yılları arasındaki iş kazası verileri kullanılarak ARIMA (1,1,0) modeli oluşturulmuş ve 2020 yılı için iş kazası sayısının 67.463 olacağı tahmin edilmiştir (Eravcı, 2018). Türkiye'de yapılan bir araştırmada, iş cinayetleri ve devamlı iş görmezlik sayıları hem ARIMA hem de YSA ile modellenmiş ve YSA' nın daha doğru tahminler sunduğu belirlenmiştir (Ergül, 2018). Bu yöntemle, geçmiş verilerin ağırlıklı ortalamaları alınarak gelecekteki iş kazası sayıları tahmin edilir. Türkiye'de iş kazalarının tahmin edilmesinde üstel düzeltme metodunun kullanıldığı çalışmalar da mevcuttur (Yağımlı & Ergin, 2017). Ayrıca olasılık dağılımları yaklaşımı ile iş

kazası verilerinin olasılık dağılımları analiz edilerek, kaza riskleri ve sıklıkları belirlenip inşaat sektörüne yönelik kaza sıklığı ve etki derecesi gibi faktörler değerlendirilerek istatistiksel modeller oluşturulmuştur (Erginel & Toptancı, 2017).

2. YÖNTEM

Bu bölümde çalışmada kullanılan yöntemler hakkında kısa bilgi verilmektedir.

2.1. ARIMA Modeli

Uygulamada karşılaşılan zaman serileri genellikle durağan olmayan serilerdir. ARMA modellerinin tahmini için öncelikle serinin durağan hale getirilmesi gerekmektedir. Serinin d. dereceden fark alma işleminin eklenmesi ile ARIMA modellerinden bahsedilmektedir. ARIMA (p, d, q) genel ifadesi z_t farkı alınmış bir zaman serisini göstermek üzere aşağıdaki gibi yazılabilir (Güriş vd., 2020);

$$z_t = \alpha_1 z_{t-1} + \alpha_2 z_{t-2} + \cdots + \alpha_p z_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_q \varepsilon_{t-q} \quad (1)$$

Δ fark alma derecesini, d fark alma derecesini göstermek üzere ARIMA (p, d, q) modeli;

$$\alpha(B)\Delta^d z_t = \theta(B)\varepsilon_t \quad (2)$$

şeklinde ifade edilmektedir.

Ayrıca mevsimsellik etkisine sahip seriler de durağan olmayan serilerdir. D mevsimsel fark alma derecesini SAR(P) mevsimsel otoregresif süreci, SMA(q) mevsimsel hareketli ortalama sürecini göstermek üzere SARIMA (p, d, q)(P, D, Q)$_s$ modeli mevsimsel otoregresif bütünleşik hareketli ortalama modelini ifade etmektedir. Model, p, AR sürecinin derecesi, q, MA sürecinin derecesi, d fark alma derecesi, P, mevsimsel AR sürecinin derecesi, D mevsimsel fark alma derecesi, Q, mevsimsel MA süreci derecesini göstermek üzere aşağıdaki gibi yazılmaktadır (Permanasari vd., 2013).

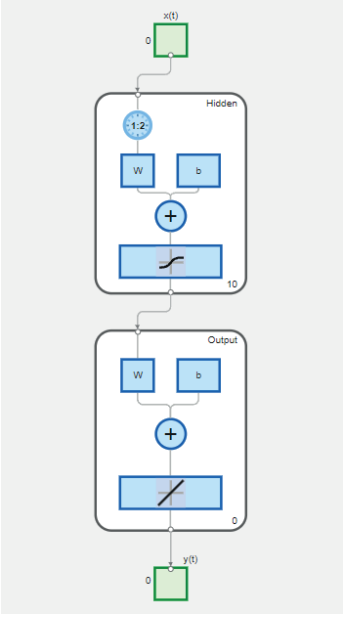
$$\varphi_p(B^S)\varphi_p(B)(1-B^S)^D(1-B)^d y_t = \Theta_Q(B^s)\theta_q(B)\varepsilon_t \quad (3)$$

2.2. Yapay Sinir Ağları

Yapay sinir ağlarının (YSA) tarihçesi, insan beyninin işleyişini anlamaya ve taklit etmeye yönelik merakla başlamıştır. Bilimsel çalışmalar, matematiksel modelleme ve bilgisayar teknolojisinin gelişimi bu süreci şekillendirmiştir. YSA' lar, biyolojik sinir sistemini esas alan bir teknoloji olarak geliştirilmiş ve günümüzde yapay zekanın en önemli bileşenlerinden biri haline gelmiştir

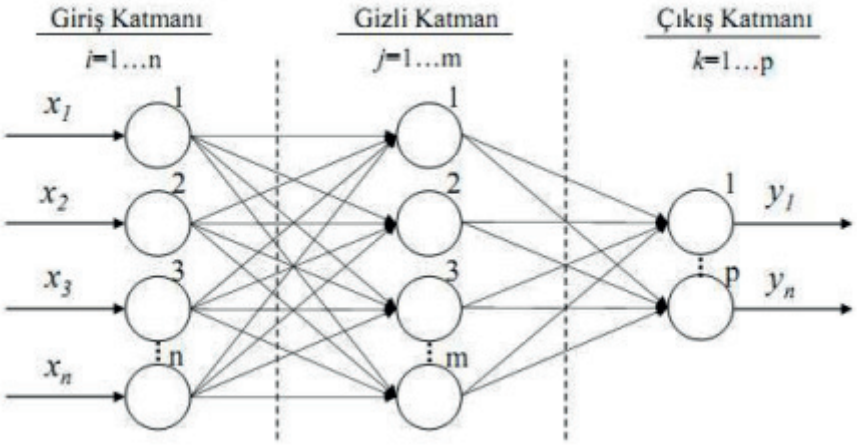
(Keskenler & Keskenler, 2017). 1943 yılında Warren McCulloch ve Walter Pitts, ilk yapay sinir ağı (YSA) modelini geliştirdi. 1956, yapay zeka ile ilgili modern araştırmaların başlangıcıdır. Günümüzde, insansı robotlar gibi gelişen teknolojilerle birlikte yapay sinir ağlarının kullanımı artmaktadır (Öztürk & Şahin, 2018). YSA, insan beyninin öğrenme ve bilgi işleme yeteneklerini taklit eden matematiksel modellerdir. İnsan beyni, nöronlar aracılığıyla bilgiyi alır, işler ve saklar. Bu süreçte, nöronlar arasındaki sinaptik bağlantılar öğrenmenin temel mekanizmasını oluşturur. İnsan beynindeki bu doğal süreç, yapay sinir ağlarının geliştirilmesinde esin kaynağı olmuştur. YSA' lar, biyolojik sinir sistemini model alan bir teknoloji olarak, öğrenme ve bilgi depolama süreçlerini simüle etmeye çalışır (Ersoy & Karal, 2012).

Yapay sinir ağları yaklaşımı, doğrusal ve doğrusal olmayan veri yapılarını öğrenme yeteneği sayesinde girdi ve çıktı değişkenleri arasındaki karmaşık ilişkileri de başarılı şek,lde analiz edebilmektedir (Ataseven, 2013). Şekil 7 yapay sinir ağı yapısını göstermektedir.



Şekil 7. Yapay Sinir Ağı Yapısı

YSA, insan beyninin fonksiyonlarından olan öğrenme ve bu yolla yeni bilgiler türetebilme özelliklerini gerçekleştirmek amacıyla geliştirilmiş modellerdir. Birden fazla yapay nöronun çeşitli şekillerde bağlandığı katmanlı bir yapıya sahiptirler. Genellikle, kendi içinde paralel 3 katmandan oluşmaktadırlar (Öztemel, 2003). Şekil 8'de 3 katmanlı bir YSA gösterilmektedir (Temür, 2013).



Şekil 8: 3 katmanlı YSA modeli

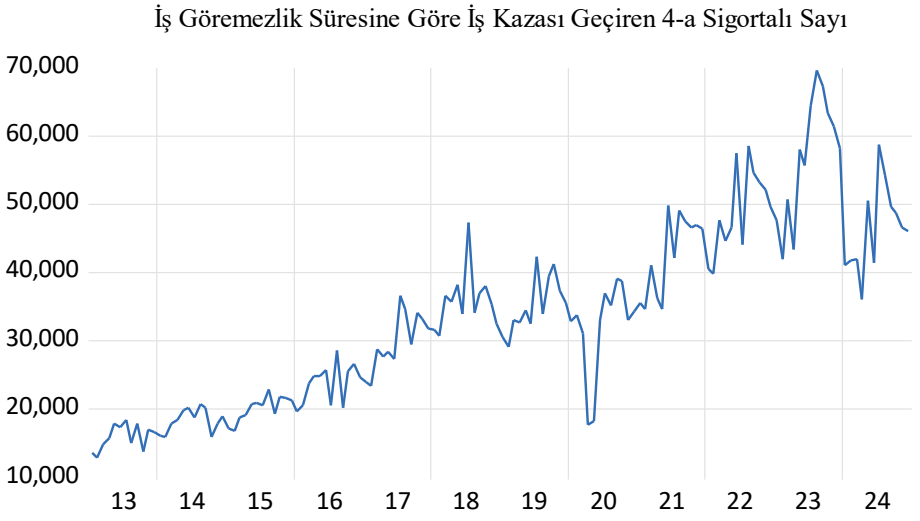
Giriş katmanındaki nöronlar, dış dünyadan aldıkları bilgileri ağa iletirler. Bazı ağlarda giriş katmanında herhangi bir bilgi işleme gerçekleştirilmez. Giriş katmanından gelen bilgiler gizli katmanda işlenir ve çıkış katmanına gönderilir. Bir YSA' da birden fazla gizli katman olabilir. Gizli katmandan gelen bilgiler işlenerek, verilen giriş değerleri için üretilmesi gereken çıkış değeri bu katmanda üretilir (Öztemel, 2003). YSA yapılarının eğitiminde çok sayıda öğrenme algoritması kullanılmaktadır. Derin öğrenme yöntemleri ise yapay sinir ağlarının gelişmiş bir biçimi olarak, özellikle büyük veri ve güçlü işlemcilerle birlikte son yıllarda ilgi odağı olmuştur. Çok katmanlı ağlar kullanılarak karmaşık desenleri öğrenir. Doğal dil, görüntü işleme, sağlık gibi bir çok alanda yaygın olarak kullanılmaktadır. Tıbbi teşhis ve görüntü analizi de bunlara örnek verilebilmektedir (Heaton, 2018).

3. ANALİZ VE BULGULAR

Bu bölümde, Türkiye'de yaşanan iş kazaları Zaman Serisi Analizi ve Yapay Sinir Ağı (YSA) ile incelenmiş ve elde edilen bulgular değerlendirilmiştir. Türkiye'deki iş kazaları sonucu oluşan sürekli iş göremezlik sayılarının tahmini için ARIMA zaman serisi analizi ve Yapay Sinir Ağları (YSA) teknikleri kullanılmıştır.

3.1. Değişkenler ve Tanımlayıcı İstatistikler

Çalışma verileri, Sosyal Güvenlik Kurumu (SGK)'dan alınmıştır. Veriler 2013-2024 yılları arasında aylık frekanstaki iş kazası sonucu iş göremezlik süresine göre iş kazası geçiren 4-a sigortalı sayısı (IKYS) verilerini kapsamaktadır (https://www.sgk.gov.tr/Istatistik/Yillik/fcd5e59b-6af9-4d90-a451-ee-7500eb1cb4, Erişim: 20.10.2025).



Şekil 9. İş Göremezlik Süresine Göre İş Kazası Geçiren Sigortalı Sayısı (IKYS) Grafiği

Şekil 9' da 2013-2024 yılları arasındaki dönemi kapsayan iş kazası verileri incelendiğinde, iş göremezlik süresine göre iş kazası geçiren 4-a sigortalı sayısının yıllar içinde genel bir artış eğilimi gösterdiği görülmektedir. 2013 yılında yaklaşık 10.000 civarında olan iş kazası sayısı, zamanla artarak 2023 yılına gelindiğinde 60.000'in üzerine çıkmıştır. Bu artış sürecinde özellikle 2018 ve 2022 yılları arasında belirgin dalgalanmalar yaşanmış, genel eğilim artma yönlü olmuştur. 2020 yılında dikkat çekici bir düşüş yaşanmış olup, bu durumun COVID-19 pandemisi nedeniyle ekonomik faaliyetlerin yavaşlaması ve birçok işyerinin kapanmasıyla ilişkili olduğu düşünülebilir. Pandemi sonrası dönemde ise iş kazası sayılarının hızlı bir şekilde artış gösterdiği, özellikle 2022 ve 2023 yıllarında en yüksek seviyelere ulaştığı gözlemlenmiştir. Bu artışın, sanayi üretimindeki artış, inşaat ve hizmet sektörlerindeki genişleme ve istihdamdaki yükselişle bağlantılı olabileceği değerlendirilebilir. 2024'te dalgalanma sürse de seviyeler hala pandemi öncesinin belirgin şekilde üzerindedir ve 35-55 bin bandında seyredip 2023'e göre bir miktar normalleşme gösteriyor.

Tablo 1. Tanımlayıcı İstatistikler

	Değişken
İstatistikler	**IKYS**
Ortalama	33860.91
Ortanca	33791.00
Minimum	12897.00
Maksimum	69514.00
Standart Sapma	13443.48
Çarpıklık	0.447149

Basıklık	2.438929
Jarque-Bera (JB) (Olasılık)	6.687413 (0.035306)

Tablo 1'de ISKGSS değişkeninin tanımlayıcı istatistikleri incelendiğinde, iş göremezlik süresine göre iş kazası geçiren sigortalı sayısı ortalamasının 33860.91 olduğu, en düşük değerin 12897.00 ve en yüksek değerin 69514.00 olduğu görülmektedir. Standart sapma 13443.48 olup, verilerin ortalamadan olan yayılımını göstermektedir. Çarpıklık değeri (0.447), dağılımın hafif sağa çarpık olduğunu, basıklık değeri (2.438) ise dağılımın normale göre basık bir dağılım sergilediğini ifade etmektedir. Jarque-Bera normallik testi sonucunda (JB=6.687, p=0.0353), serinin %5 anlamlılık düzeyinde normal dağılımdan sapma gösterdiği görülmektedir.

3.2. SARIMA Modeli Bulguları

Model tahmini öncesinde serinin durağanlığı için Birim Kök Testi sonuçlarına yer verilmiştir.

3.2.1. Birim Kök Testleri

Bu çalışmada analiz edilen "İş Göremezlik Süresine Göre İş Kazası Geçiren 4-a Sigortalı Sayısı" değişkeni (İKYS) değişkeninin durağan olup olmadığını belirlemek amacıyla sırasıyla Augmented Dickey-Fuller (ADF) testi, Phillips-Perron (PP) testi ve Kwiatkowski-Phillips-Schmidt-Shin (KPSS) testi uygulanmıştır.

Tablo 2. İKYS İçin Birim Kök Testi Sonuçları

Test	**Test İstatistiği**
ADF	-1.317481 p-Değ. (0.6202)
PP	-2.186374 p-Değ. (0.2122)
KPSS	1.417261 Kritik Değ. (0.4630)

Tablo 2 sonuçlarına göre ADF, PP ve KPSS test sonuçları % 5 anlamlılık düzeyinde serinin durağan olmadığını göstermektedir. Tablo 3' de ilk farkları alınmış İKYS Birim Kök Test bulgularına yer verilmiştir.

Tablo 3. İlk Farkı Alınan İKYS Serisinin Birim Kök Testi Sonuçları

Test	Test İstatistiği
ADF	-3.245977 p-Değ. (0,0196)
PP	-19.11786 p-Değ. (0,000)
KPSS	0.081329 Kritik Değ. (0.4630)

Tablo 3' den serinin ilk farkına uygulanan birim kök testleri sonuçlarına göre %5 anlamlılık düzeyinde durağan olduğunu söylemek mümkündür.

Date: 12/07/25 Time: 19:21
Sample (adjusted): 2013M02 2024M12
Included observations: 143 after adjustments

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	-0.435	-0.435	27.596	0.000
		2	0.171	-0.022	31.886	0.000
		3	0.034	0.124	32.061	0.000
		4	-0.117	-0.067	34.095	0.000
		5	-0.024	-0.152	34.180	0.000
		6	-0.166	-0.280	38.357	0.000
		7	-0.034	-0.238	38.533	0.000
		8	-0.105	-0.247	40.222	0.000
		9	0.191	0.104	45.854	0.000
		10	-0.167	-0.070	50.212	0.000
		11	0.012	-0.301	50.233	0.000
		12	0.445	0.348	81.525	0.000
		13	-0.349	-0.011	100.99	0.000
		14	0.243	-0.082	110.51	0.000
		15	-0.028	0.086	110.64	0.000
		16	-0.037	0.083	110.86	0.000
		17	-0.054	-0.092	111.34	0.000
		18	-0.043	0.024	111.65	0.000
		19	-0.119	-0.030	114.02	0.000
		20	-0.009	-0.108	114.03	0.000
		21	0.202	0.199	120.96	0.000
		22	-0.373	-0.142	144.80	0.000
		23	0.384	0.059	170.22	0.000
		24	-0.062	-0.070	170.90	0.000
		25	-0.057	-0.001	171.47	0.000
		26	0.122	-0.056	174.11	0.000
		27	0.010	0.124	174.13	0.000
		28	-0.059	-0.027	174.75	0.000
		29	-0.013	-0.064	174.79	0.000
		30	-0.065	-0.151	175.57	0.000
		31	-0.096	0.024	177.29	0.000
		32	0.066	-0.092	178.12	0.000
		33	0.012	-0.026	178.14	0.000
		34	-0.230	-0.056	188.18	0.000
		35	0.424	0.043	222.72	0.000
		36	-0.229	-0.026	232.87	0.000

Şekil 10. Durağan IKYS serisinin ACF ve PACF Grafikleri

Birim Kök Test bulgularına göre; birinci farkı alınarak ($d=1$) durağan hale getirilen IKYS serisi için uygun ARIMA model yapısını belirlemek amacıyla Şekil 10' da verilen otokorelasyon (ACF) ve kısmi otokorelasyon (PACF) grafikleri incelenmiştir. Elde edilen korelogram grafiğinde, serinin birinci gecikmesinde (Lag 1) istatistiksel olarak anlamlı bir ilişki olduğu ve kat-

sayıların güven sınırlarının dışına taştığı görülmüştür. Bu durum, bir önceki dönemdeki yaralanma sayılarının mevcut dönemi açıklamakta etkili olduğunu ve serinin otoregresif (AR) bir yapı sergilediğini göstermektedir. Ayrıca, veri setinin aylık frekansa sahip olması nedeniyle mevsimsellik etkisi incelenmiş ve ACF grafiğinin 12. gecikmesinde (Lag 12) anlamlı bir ilişki tespit edilmiştir. Bu durumda iş kazası ve yaralanma verilerinde yıllık periyotlarla tekrar eden güçlü bir mevsimsel döngünün varlığını kanıtlamaktadır. Bu bulgular ışığında serinin durağanlık derecesi (d=1) olarak alınmış, kısa dönemli ilişkileri açıklamak için otoregresif bileşen AR(1) ve mevsimsel etkileri modellemek için mevsimsel hareketli ortalama SMA(12) terimlerinin modele dahil edilmesi ile SARIMA modeli tahmini önerilmiştir. Model tahmini sonuçları Tablo 4' de verilmiştir.

Tablo 4. SARIMA(1,1,0)(0,0,1)$_{12}$ Modeli Katsayı Tahminleri

Değişken	**Katsayı**	**Standart Hata**	**Test İstatistiği**	**P-Değeri**
Sabit	155.8629	404.7779	0.385058	0.7008
AR(1)	-0.406647	0.073801	-5.510038	0.0000
MA(12)	0.575671	0.069512	8.281605	0.0000
Kriterler	**AIC**	**SC**	**HQ**	**LogL**
	19.56306	19.64594	19.59674	-1394.759

Tablo 4 tahmin sonuçları model parametrelerinin sabit hariç % 5 anlamlılık düzeyinde istatistiksel anlamlı olduğunu göstermektedir. Tablo 5, SARIMA(1,1,0)(0,0,1)$_{12}$ modelinin dönem içi tahmin değerlerini göstermektedir.

Tablo 5. Gerçek Değerler ve SARIMA(1,1,0)(0,0,1)$_{12}$ Tahmin Değerleri Karşılaştırması

Tarih	**Gerçek Değer**	**Tahmin Değeri**	**Hata**
2023-6	55706	56926	-1220
2023-7	64432	53319	11113
2023-8	69514	65013	4501
2023-9	67326	68685	-1359
2023-10	63406	66951	-3545
2023-11	61323	63333	-2010
2023-12	58145	60738	-2593
2024-1	41867	45319	-3452
2024-2	41889	44321	-2432
2024-3	36143	40990	-4847
2024-4	50518	45063	5455
2024-5	41389	44189	-2800
2024-6	58670	51718	6952

2024-7	54302	54453	-151
2024-8	49559	55515	-5956
2024-9	48810	49666	-856
2024-10	46552	48177	-1625
2024-11	46138	46197	-59
2024-12	41867	45319	-3452

Tablo 6. SARIMA$(1,1,0)(0,0,1)_{12}$ Performans Değerleri

Model	MSE	RMSE
SARIMA$(1,1,0)(0,0,1)_{12}$	16956976.873225	4117.885

Tablo 6'da, SARIMA$(1,1,0)(0,0,1)_{12}$ modelinin hata metriklerini değerlendirmek amacıyla MSE (Ortalama Kare Hata) ve RMSE (Kök Ortalama Kare Hata) kriterleri elde edilmiştir. Özellikle mevsimsel etkilerin modele dahil edilmesinin tahmin başarısını artırdığını dolayısıyla iş kazası verileri gibi karmaşık yapıdaki verilerin tahmin edilmesinde güvenilir bir model yaklaşımı sağlayabileceğini söylemek mümkündür.

3.3. YSA Modeli Bulguları

Bu bölümde İKYS serisi için Yapay Sinir Ağları (YSA) yaklaşımı tahmin sonuçlarına yer verilmiştir. Verinin %75' i eğitim, %15' u doğrulama ve %15' i test için ayrılmıştır. Şekil 11, bir girdi, bir çıktı ve 10 gizli nöronlu ağ yapısı tahmin ekranı verilmiştir.

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	11	1000
Elapsed Time	-	00:00:00	-
Performance	1.7e+09	1.44e+07	0
Gradient	3.42e+09	2.2e+07	1e-07
Mu	0.001	1e+05	1e+10
Validation Checks	0	6	6

Şekil 11. YSA Tahmin Ekranı

Modelin eğitimi için Levenberg–Marquardt algoritması uygulanmış ve model eğitimi 11 epoch sonunda tamamlanmıştır. Başlangıçta 1.7×10^{9} olan performans (MSE) değeri, eğitim sonunda 1.44×10^{7} seviyesine düşmüş, bu durum modelin hata değerinde belirgin bir iyileşme sağladığını göstermiştir. Gradyan değeri 3.42×10^{9}'dan 2.2×10^{7}'ye gerilemiş olmakla birlikte, hedeflenen 1×10^{-7} seviyesine ulaşılamamıştır. Doğrulama kontrolü sayısının 6'ya ulaşması nedeniyle eğitim süreci erken durdurma (early stopping) kriterine göre sonlandırılmıştır.

Tablo 7. YSA Modeli Performans Değerleri

YÖNTEM	EĞİTİM ALGORİTMASI	R	MSE	RMSE
YSA	Levenberg-Marquardt	0.9231	3.2049e+07	5661.1

Yapay Sinir Ağı modeli, Levenberg-Marquardt algoritmasıyla eğitilmiş ve modelin R değeri 0.9231 olarak elde edilmiştir. Bu sonuçlar, modelin güçlü bir tahmin performansı sergilediğini göstermektedir. YSA modeline göre IKYS serisinin gerçek ve tahmin değerleri Tablo 9' da verilmiştir.

Tablo 8. Gerçek Değerler ve YSA Tahmin Değerleri Karşılaştırması

Tarih	Gerçek Değer	Tahmin Değeri	Hata
2023-6	55706	54951	755
2023-7	64432	47637	16795
2023-8	69514	63691	5823
2023-9	67326	82213	-14887
2023-10	63406	65400	-1994
2023-11	61323	53178	8145
2023-12	58145	50740	7405
2024-1	41093	48232	-7139
2024-2	41867	24446	17421
2024-3	41889	43075	-1186
2024-4	36143	43520	-7377
2024-5	50518	40494	10024
2024-6	41389	46608	-5219
2024-7	58670	44824	13846
2024-8	54302	56184	-1882
2024-9	49559	47137	2422
2024-10	48810	46764	2046
2024-11	46552	47078	-526
2024-12	46138	46616	-478

3.4. SARIMA ve YSA Model Tahmin Performanslarının Karşılaştırılması

Tablo 9' da IKYS serisi için kullanılan tahmin modellerinin performans karşılaştırmasını göstermektedir.

Tablo 9. YSA ve SARIMA(1,1,0)(0,0,1)$_{12}$ Model Tahmin Performans Değerleri

Model	MSE	RMSE
SARIMA(1,1,0)(0,0,1)$_{12}$	16956976.873	4117.885
YSA (Levenberg-Marquardt)	32049000	5661.100

Tablo 9'daki MSE ve RMSE hata metriklerine göre SARIMA(1,1,0)(0,0,1)$_{12}$ modelinin tahmin performansının, YSA modeline göre daha iyi olduğunu söylemek mümkündür. Buna göre iş kazası sonucu iş göremezlik süresine göre iş kazası geçiren 4-a sigortalı sayısı (IKYS) değişkeninin en iyi performansı gösteren SARIMA(1,1,0)(0,0,1)$_{12}$ modeline göre 2025 yılı için öngörü değerleri Tablo 10' da verilmiştir.

Tablo 10. SARIMA(1,1,0)(0,0,1)$_{12}$ Modeli ile IKYS için 2025 Yılı Öngörü Değerleri

Tarih	Öngörü Değerleri
2025-1	35443
2025-2	38024
2025-3	35794
2025-4	34130
2025-5	38166
2025-6	35132
2025-7	40587
2025-8	38501
2025-9	36140
2025-10	36827
2025-11	35831
2025-12	36422

Tablo 10'da SARIMA(1,1,0)(0,0,1)$_{12}$ modelinin öngörülerine göre 2025 yılında belirli bir mevsimsel patern izleyerek dalgalı bir seyir izleyeceği tahmin edilmektedir. Yıl içerisindeki en yüksek vaka sayısının 40587 ile Temmuz ayında gerçekleşmesi beklenirken, en düşük vaka sayısının 34130 ile Nisan ayında olacağı öngörülmektedir. Tahminlere göre Mayıs, Temmuz ve Ağustos aylarındaki yüksek değerler, iş kazalarının yaz aylarında artış eğilimine girdiğini işaret etmektedir. Bu durum, geçmiş verilerde tespit edilen 12 aylık mevsimsel döngünün 2025 yılında da belirleyici faktör olacağını göstermektedir.

4. SONUÇ VE ÖNERİLER

Bu çalışmada, 2013-2024 yılları arasında Türkiye' de iş kazası sonucu iş göremezlik süresine göre iş kazası geçiren 4-a sigortalı sayısı (IKYS) verilerinin tahmini için mevsimsel ARIMA(SARIMA) ve YSA modeli yaklaşımı kullanılarak tahmin performansları karşılaştırılmıştır. Bulgular hata metriklerine göre, IKYS değişkeninin tahmini için SARIMA$(1,1,0)(0,0,1)_{12}$ modelinin en iyi performans gösterdiğini destekler niteliktedir. Modelin 2025 yılı öngörüleri değerlendirildiğinde elde edilen bu sonuçlar iş sağlığı ve güvenliği önlemlerinin ve denetimlerinin, kaza riskinin yüksek olarak tahmin edildiği Temmuz ayı için özellikle artırılması gerektiğini söylemek mümkündür.

İş kazalarına ilişkin verilerin içerdiği mevsimsel döngüler ve trend bileşenleri, SARIMA modelinin yapısal özellikleri tarafından başarılı bir şekilde modellenebilmektedir. YSA modelleri ise doğrusal olmayan ilişkileri yakalayabilme kapasitesine sahip olsa da bu çalışmada verinin belirgin mevsimsel karakteristiğini modellemede SARIMA modeli kadar hassas sonuçlar üretememiştir.

Sonuç olarak incelenen zaman serisi verileri için SARIMA$(1,1,0)(0,0,1)_{12}$ modeli, daha düşük hata metrikleri ve veri setinin içsel dinamiklerine daha iyi uyum sağlaması nedeniyle, YSA modeline tercih edilerek en uygun tahmin yöntemi olarak belirlenmiştir. Bu durum, karmaşık yapay zeka algoritmalarının her zaman en iyi sonucu vermeyebileceğini, verinin yapısına uygun istatistiksel modellerin de yüksek doğruluk sağlayabileceğini ortaya koymaktadır. Çalışmanın geliştirilebilir yönü olarak, makine öğrenmesi, derin öğrenme yöntemleri ile zaman serisi modellerinin hibrit olarak kullanıldığı farklı tahmin yaklaşımlarının performans karşılaştırılması sonraki çalışmalar için değerlendirilmektedir.

5. KAYNAKÇA

Akay, A. O., Akgül, M., Esin, A. I., Demir, M., Şentürk, N., & Öztürk, T. (2021). Evaluation of occupational accidents in forestry in Europe and Turkey by k-means clustering analysis. *Turkish Journal of Agriculture and Forestry*, *45*(4), 495-509.

Ataseven, B. (2013). Yapay sinir ağlari ile öngörü modellemesi. *Öneri Dergisi*, *10*(39), 101-115.

Bayraktaroğlu, S., Aras, M. & Atay, E. (2018). Çalışanlarda İş Güvenliği ve İş Kazası Algısı: Mavi Yakalılar Üzerine Bir Araştırma. *Uluslararası Yönetim ve Sosyal Araştırmalar Dergisi, 5(9),* 1-15.

Bekar, İ., Oruç, D. & Bekar, E. (2017). İş kazası ve meslek hastalıklarının maliyeti (2005-2014). *Uluslararası Ekonomik Araştırmalar Dergisi, 3(3), 479-489.*

Bilim, A., & Çelik, O. N. (2018). Türkiye'deki İnşaat Sektöründe Meydana Gelen İş Kazalarının Genel Değerlendirmesi. *Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi, 7(2),* 725-731.

Ceylan, H. (2011). Türkiye'deki iş kazalarının genel görünümü ve gelişmiş ülkelerle kıyaslanması. *International Journal of Engineering Research and Development*, *3*(2), 18-24.

Ceylan, H. (2014). An artificial neural networks approach to estimate occupational accident: a national perspective for Turkey. *Mathematical problems in engineering*, *2014*(1), 756326.

Ceylan, H. (2021). Türkiye'de Meydana Gelen Ölümlü İş Kazaları. *İSG Akademik – OHS Academic 3(1),* 1-13.

Çalış, S., & Büyükakıncı, B. Y. (2021). Türkiye'nin iş kazaları açısından durumu: ILOSTAT ve SGK verileri karşılaştırması. *Afyon Kocatepe Üniversitesi Sosyal Bilimler Dergisi*, *23*(2), 574-585.

Çavuş, A. & Taçgın E. (2016). Türkiye'de inşaat sektöründeki iş kazalarının sınıflandırılarak nedenlerinin incelenmesi. *Academic Platform-Journal of Engineering and Science, 4(2).*

Dyreborg, J., Lipscomb, H. J., Nielsen, K., Törner, M., Rasmussen, K., Frydendall, K. B., & Kines, P. (2022). Safety interventions for the prevention of accidents at work: A systematic review. *Campbell systematic reviews*, *18*(2), e1234. 10.1002/cl2.1234.

Eravcı, D. B. (2018). İş Kazalarının Box-Jenkins ARIMA Tekniği Kullanılarak Modellenmesi. *Çalışma İlişkileri Dergisi*, 9(1), 58-71.

Erginel, N., & Toptancı, Ş. (2017). İş Kazası Verilerinin Olasılık Dağılımları İle Modellenmesi. *Mühendislik Bilimleri ve Tasarım Dergisi*, *5*, 201-212.

Ergül, B. (2018). Türkiye'deki İş Kazalarının Zaman Serisi Analiz Teknikleri ve Yapay Sinir Ağları Tekniği İle İncelenmesi. *Karaelmas Journal of Occupational Health and Safety*, *2*(2), 63-74.

Ersoy, E., & Karal, Ö. (2012). Yapay sinir ağları ve insan beyni. *İnsan ve Toplum Bilimleri Araştırmaları Dergisi*, *1*(2), 188-205.

Güğercin, Ö., & Baytorun, A. N. (2018). Tarımda iş kazaları ve gerekli önlemler. *Çukurova Tarım ve Gıda Bilimleri Dergisi*, *33*(2), 157-168.

Gül, M., Güneri, A. F., Yılmaz, F., & Çelebi, O. (2016). Analysis of the relation between the characteristics of workers and occupational accidents using data mining. *The Turkish Journal of Occupational/Environmental Medicine and Safety, 1(4),*102-118.

Güriş S.; Akay E. Ç. & Güriş, B. (2020). *R ile Temel Ekonometri, Der Yayınları, İstanbul.*

Heaton, J. (2018). Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618. *Genetic programming and evolvable machines*, *19*(1), 305-307.

Jacinto, C., & Aspinwall, E. (2003). Work accidents investigation technique (WAIT)–Part I. *Safety Science Monitor*, *7*(1), 1-17.

Karadeniz, O. (2012). Dünya'da ve Türkiye'de iş kazaları ve meslek hastalıkları ve sosyal koruma yetersizliği. *Çalışma ve Toplum*, *3*(34), 15-73.

Keskenler, M. F., & Keskenler, E. F. (2017). Geçmişten günümüze yapay sinir ağları ve tarihçesi. *Takvim-i Vekayi*, *5*(2), 8-18.

Mammadov, A. (2021). *Derivation of Prescriptive Accident Prevention Model From Predictive Models Using Ml Algorithms* (Master's thesis, Middle East Technical University (Turkey)).

Mohammadfam, I., Soltanzadeh, A., Moghimbeigi, A., & Savareh, B. A. (2015). Use of artificial neural networks (ANNs) for the analysis and modeling of factors that affect occupational injuries in large construction industries. *Electronic physician*, *7*(7), 1515.

Nedenisg (2021). *İş kazaları ve meslek hastalıkları istatistiği 2020.* [Erişim: 17.03.2025, https://nedenisguvenligi.com/is-kazalari-ve-meslek-hastaliklari-istatistigi-2020/]

Öçal, M., & Çiçek, Ö. (2017). Türkiye Ve Avrupa Birliği'nde İş Kazası Verilerinin Karşılaştırmalı Analizi. *Hak İş Uluslararası Emek ve Toplum Dergisi*, *6*(16), 616-637.

Özdemir, M. (2024). Work Accidents, Occupational Diseases, and Lost Workdays in Türkiye's Forestry Sector: Increasing Risks and Improvement Proposals for the 2019-2023 Period. *Journal of Apitherapy and Nature*, *7*(2), 141-153.

Öztemel, E., Yapay sinir ağları. *Papatya Yayıncılık,* İstanbul, 2013.

Öztürk, K., & Şahin, M. E. (2018). Yapay sinir ağları ve yapay zekâ'ya genel bir bakış. *Takvim-i Vekayi*, *6*(2), 25-36.

Şen, M., Dursun, S., & Murat, G. (2018). Türkiye'de iş kazaları: Avrupa birliği ülkeleri bağlamında bir değerlendirme. *OPUS International Journal of Society Researches, 9(16), 1167-1190.*

Taylan, M. (2008). Tersanelerde Meydana Gelen İş Kazaları Ve İş Güvenliği. *Gemi İnşaatı Ve Deniz Teknolojisi Teknik Kongresi, 270-281.*

Permanasari, A. & Hidayah, I.& Bustoni, I. (2013). SARIMA (Seasonal ARIMA) imp-

lementation on time series to forecast the number of Malaria incidence. 203-207. 10.1109/ICITEED.2013.6676239.

Ünal, H. G., Yaman K. & Gök A. (2008). Türkiye'de tarımsal iş kazaları ve meslek hastalıklarının maliyeti üzerine bir araştırma. *Journal of Agricultural Sciences, 14(04),* 428-435.

Yağımlı, M., & Ergin, H. (2017). Türkiye'de iş kazalarının üssel düzeltme metodu ile tahmin edilmesi. *Marmara Fen Bilimleri Dergisi*, *29*(4), 118-123.

Yardım, N., Çipil, Z., Vardar, C., & Mollahaliloğlu, S. (2007). Türkiye iş kazaları ve meslek hastalıkları: 2000-2005 yılları ölüm hızları. *Dicle Tip Dergisi*, *34*(4), 264-271.

Bölüm 8

KALP HASTALIKLARININ TAHMİNİNDE SAĞLIK VERİLERİNE DAYALI MAKİNE ÖĞRENİMİ YAKLAŞIMI

Erol TERZİ[1]

Meltem FİTOZ[2]

Mehmet Şirin ATEŞ[3]

1Prof.Dr. Erol TERZİ, Ondokuz Mayıs Üniversitesi, İstatistik Bölümü
eroltrz@omu.edu.tr
ORCID ID: 0000-0002-2309-827X

2 Meltem FİTÖZ, Ondokuz Mayıs Üniversitesi, İstatistik Bölümü
meltem.yaran@gmail.com
ORCID ID: 0009-0008-1397-2913

3Arş.Gör. Mehmet Şirin ATEŞ, Ondokuz Mayıs Üniversitesi, İstatistik Bölümü
mehmet.ates@omu.edu.tr
ORCID ID:0000-0001-9904-6380

1. GİRİŞ

Makine öğrenmesi algoritmaları, günümüz veri bilimi uygulamalarının temelini oluşturmaktadır. Bu algoritmalar arasında, hem sınıflandırma hem de regresyon görevleri için kullanılabilen karar ağaçları (decision trees), açıklanabilirlikleri ve sezgisel yapıları sayesinde öne çıkmaktadır. Karar ağaçları, veriyi özelliklerine göre dallara ayırarak hiyerarşik bir yapı içerisinde karar kuralları oluşturur ve bu yönüyle özellikle yorumlamaya açık, kullanıcı dostu modeller sunar (Quinlan, 1986). Ağaç yapısının kök, düğüm ve yapraklardan oluşması, karar alma sürecinin görsel ve kavramsal olarak kolay takip edilmesini sağlar. Karar ağaçlarının basit yapısına rağmen, karmaşık veri kümelerinde etkili tahminler sunabilmesi; tıp, finans, eğitim ve pazarlama gibi farklı disiplinlerde yoğun kullanımına olanak sağlamıştır (Safavian ve Landgrebe, 1991). Bununla birlikte, aşırı öğrenme (overfitting) gibi sınırlamaları da beraberinde getirmesi nedeniyle, budama (pruning) ve topluluk yöntemleri (örneğin Random Forest) gibi geliştirmelere ihtiyaç duymaktadır (Breiman vd. 1984). Bu çalışma, karar ağaçlarının temel ilkelerini, avantajlarını, sınırlılıklarını ve çeşitli uygulama alanlarındaki etkinliğini ele almayı amaçlamaktadır.

Makine öğrenimi (ML) algoritmaları, büyük ve karmaşık veri kümeleri üzerinde desenleri tanımlama ve tahminler üretme konusunda etkili araçlar sunmaktadır. Özellikle sağlık alanında, hasta geçmişi, demografik bilgiler, biyokimyasal veriler ve görüntüleme sonuçları gibi çok boyutlu verilerin değerlendirilmesinde makine öğrenimi yöntemleri giderek daha fazla kullanılmaktadır (Shickel vd. 2018). Lojistik regresyon, karar ağaçları, destek vektör makineleri (SVM), k-en yakın komşu (KNN) ve rastgele ormanlar (Random Forest) gibi algoritmalar, kalp hastalığı tahmini konusunda sıkça başvurulan yöntemler arasında yer almaktadır (Khan vd. 2020).

Bu çalışmanın amacı, sağlık verilerini kullanarak kalp hastalığı riskini tahmin edebilen bir makine öğrenimi modeli geliştirmektir. Bu doğrultuda, çeşitli ML algoritmaları karşılaştırılarak en yüksek doğruluk oranına sahip model belirlenmeye çalışılacaktır. Ayrıca, modelin yorumlanabilirliğini artırmak amacıyla özellik önem düzeyleri de analiz edilecektir. Çalışma,

hem klinik karar destek sistemlerinin geliştirilmesine katkı sunmayı hem de sağlık hizmetlerinde erken müdahale süreçlerini desteklemeyi hedeflemektedir.

2. YAPAY ZEKA MODELLERİ

Yapay zekâ (YZ), makine öğrenmesi kullanılmaksızın dahi, sistemlerin tahmin yapma, karar verme ve problem çözme gibi bilişsel işlevleri yerine getirebilmesini sağlayan teknolojik bir yaklaşımdır. Bu tür sistemler, öğrenme yoluyla çeşitli kurallar geliştirir, mantıksal çıkarımlar üretir ve kendini sürekli geliştirerek çıktı üretme kapasitesine sahiptir (Russell ve Norvig, 2020).

Yapay zekâ kavramı ilk kez 1956 yılında, ABD'nin New Hampshire eyaletinde Dartmouth Koleji'nde gerçekleştirilen akademik bir çalıştayda resmen tanımlanmıştır (McCarthy vd. 1956). Bu çalıştay, yapay zekânın bilimsel bir disiplin olarak doğuşunu temsil eder. Alan Turing'in 1950 yılında yayımladığı “Computing Machinery and Intelligence” başlıklı makalesi ise, bir makinenin düşünme yeteneğini ölçmeyi amaçlayan Turing Testi'ni tanımlayarak bu alanın kuramsal temelini oluşturmuştur (Turing, 1950).

Soğuk Savaş döneminde, özellikle Amerika Birleşik Devletleri'nin Sovyetler Birliği'ne karşı istihbarat üstünlüğü elde etme çabaları kapsamında, yapay zekâya yönelik araştırmalar stratejik bir öncelik kazanmış, ancak 1970’li yılların başlarında yeterli ilerleme kaydedilemediği için bu alandaki fonlamalar büyük ölçüde azaltılmıştır (Crevier, 1993).

Yapay zekâya olan ilgi, 1990'lı yıllarda özel sektörün teknolojiye yaptığı yatırımlar sayesinde yeniden canlanmıştır. 1997 yılında IBM tarafından geliştirilen Deep Blue adlı satranç bilgisayarı, dünya satranç şampiyonu Garry Kasparov’u mağlup ederek hem kamuoyunda büyük yankı uyandırmış hem de YZ sistemlerinin karar verme yeteneklerinin gücünü gözler önüne sermiştir.

Son yıllarda ise derin öğrenme algoritmalarının gelişmesiyle yapay zekânın güvenlik alanındaki potansiyeli artmıştır. Google DeepMind ekibi tarafından 2016 yılında geliştirilen bir sinir ağı, bilgisayarlar arası şifreli iletişim kurmayı öğrenmiş ve böylece güvenlik uygulamalarında YZ kullanımının önü açılmıştır (Abadi ve Andersen, 2016).

2.1. Denetimli Öğrenme

Denetimli öğrenme, giriş (özellik) verilerinin yanı sıra doğru sonuç etiketlerinin de bulunduğu veri kümeleri ile eğitilen algoritmalardır. Bu yöntem, sınıflandırma ve regresyon gibi görevlerde kullanılır. Örneğin, spam e-posta filtreleme sistemlerinde gelen iletilerin "spam" veya "normal" olarak etiketlenmesi bu tür bir öğrenmeye örnektir. Yaygın modeller arasında lojistik regresyon, karar ağaçları ve destek vektör makineleri yer alır (Abadi ve Andersen, 2016).

2.2. Denetimsiz Öğrenme

Denetimsiz öğrenme, etiketlenmemiş verilerde gizli yapıları bulmayı amaçlar. En bilinen uygulamalarından biri, müşteri segmentasyonu için kullanılan kümeleme algoritmalarıdır. K-Ortalamalar (K-Means) ve Ana Bileşenler Analizi (PCA), bu kategoride yer alan başlıca yöntemlerdir. Bu tür öğrenme, özellikle büyük veri kümeleri üzerinde desen keşfetmek için tercih edilir (NVIDIA Blog, 2024).

2.3. Pekiştirmeli öğrenme

Pekiştirmeli öğrenme, bir ajanın çevresiyle etkileşimi sonucu ödül veya ceza alarak politika öğrenmesi sürecidir. Ajanın amacı, uzun vadede en yüksek toplam ödülü elde etmektir. Bu yöntem, özellikle oyun oynayan yapay zekâlar ve otonom sistemlerde etkilidir. Örneğin, DeepMind tarafından geliştirilen MuZero algoritması, herhangi bir oyun kuralını bilmeden öğrenme yeteneği ile öne çıkmıştır (Schrittwieser vd., 2020).

Yapay zekâ modelleri, farklı veri türleri ve uygulama gereksinimlerine göre geliştirilmiştir. Denetimli öğrenme, etiketli veri gerektirirken yüksek doğruluk sunar; denetimsiz öğrenme, büyük veri kümelerinden bilgi çıkarımına uygundur; pekiştirmeli öğrenme ise çevresel etkileşimlerle öğrenmeyi mümkün kılar. Bu üç model grubu, yapay zekâ teknolojilerinin temel yapı taşlarıdır ve sağlık, finans, savunma ve ulaşım gibi birçok sektörde aktif olarak kullanılmaktadır (Russell ve Norvig, 2020).

“Developing AI models” ifadesinin açılımı, yani Türkçe karşılığı ve detaylı açıklaması şu şekilde olabilir:

“Developing AI models” → “Yapay Zeka Modelleri Geliştirme”

Bu, yapay zekâ (AI) sistemleri oluşturmak için veri ve algoritmalar kullanarak modeller oluşturma sürecidir.

Açılımı adım adım:

Developing (Geliştirme): Yeni bir şey inşa etmek, oluşturmak veya iyileştirmek. Burada model oluşturma ve optimize etme sürecini ifade eder.

AI (Artificial Intelligence) (Yapay Zekâ): İnsan benzeri zekâ davranışlarını taklit eden bilgisayar sistemleri.

Modes (Modeller): Verilerden öğrenip tahmin veya karar veren algoritmik yapılar.

Veri ve algoritmalar kullanarak, belirli bir problemi çözmek için yapay zeka tabanlı tahmin veya karar sistemleri oluşturma süreci.

1. Problemin Tanımlanması: Hangi problemi çözmek istediğinizi netleştirmek. Örneğin: Kalp hastalığını tahmin etmek, müşteri davranışını analiz etmek, görüntü tanıma yapmak gibi.

2. Veri Toplama: Modeli eğitmek için gereken verilerin toplanması. Veriler genellikle farklı kaynaklardan gelir: veri tabanları, sensörler, internet, kullanıcı girdileri vb.

3. Veri Ön İşleme: Eksik verilerin doldurulması, hataların düzeltilmesi. Verilerin temizlenmesi, uygun formata getirilmesi. Kategorik verilerin sayısallaştırılması (encoding). Özelliklerin ölçeklendirilmesi (scaling) ve normalizasyonu.

4. Özellik Seçimi ve Mühendisliği: Model için önemli özelliklerin (feature) seçilmesi. Yeni anlamlı özelliklerin oluşturulması. Gereksiz veya zarar veren özelliklerin çıkarılması.

5. Model Seçimi: Problemin türüne göre uygun algoritmanın belirlenmesi (örneğin; Random Forest, SVM, Neural Networks). Farklı modellerin denenmesi.

6. Model Eğitimi (Training): Verileri kullanarak modelin öğrenmesini sağlamak. Parametrelerin optimize edilmesi (hyperparameter tuning).

7. Model Değerlendirme: Test verileri üzerinde model performansının ölçülmesi. Doğruluk, hatırlama (recall), kesinlik (precision), F1 skor gibi metriklerin hesaplanması. Gerekirse modelin yeniden eğitilmesi veya iyileştirilmesi.

8. Model Testi ve Validasyonu: Modelin yeni, görülmemiş veriler üzerindeki başarısını doğrulama. Overfitting (aşırı uyum) veya underfitting (yetersiz uyum) kontrolü.

9. Modelin Uygulanması: Modelin gerçek dünyada kullanıma alınması. Sistemlere entegre edilmesi.

10. Modelin İzlenmesi ve Güncellenmesi: Modelin performansını zaman içinde takip etmek. Veriler değiştikçe veya yeni bilgiler geldikçe modeli güncellemek.

2.4. Random Forest

Random Forest, sınıflandırma ve regresyon problemlerinde yaygın olarak kullanılan güçlü bir ansamble (topluluk) öğrenme yöntemidir. Temel olarak, birçok karar ağacının (decision tree) bir araya getirilmesiyle oluşturulan bir modeldir. Her bir karar ağacı, eğitim verisinin rastgele alt örnekleri ve rastgele seçilen özellikler kullanılarak eğitilir. Bu yöntem, modelin aşırı öğrenme (overfitting) yapma riskini azaltır ve genel performansını artırır (Breiman, 2001).

Random Forest, farklı ağaçların sonuçlarını çoğunluk oyu (sınıflandırmada) veya ortalama (regresyonda) ile birleştirerek karar verir. Bu sayede, tek bir karar ağacının sahip olduğu yüksek varyans sorunu önemli ölçüde azaltılır. Ayrıca, model değişkenlerin önemini (feature importance) ölçme imkânı da sağlar, bu da özellikle veri keşfi ve yorumlama süreçlerinde büyük avantaj sağlar (Liaw ve Wiener, 2002).

Random Forest algoritması, biyoinformatik, finansal tahminler, tıp ve görüntü işleme gibi birçok alanda başarıyla uygulanmaktadır. Güçlü ve esnek yapısı, farklı veri setlerine uyum sağlamasını kolaylaştırır (Cutler vd., 2007).

2.5. Lojistik Regresyon

Lojistik regresyon, özellikle sınıflandırma problemlerinde kullanılan temel ve yaygın bir istatistiksel modeldir. Amaç, bağımsız değişkenlerin doğrusal kombinasyonuna dayanarak bir olayın olasılığını tahmin etmektir. En çok ikili sınıflandırma (binary classification) için kullanılır; örneğin, bir hastanın hastalığa yakalanıp yakalanmayacağının tahmini gibi (Hosmer vd. 2013).

Model, çıktı olarak 0 ile 1 arasında değişen bir olasılık değeri üretir ve bu değer bir eşik (genellikle 0.5) kullanılarak sınıflandırmaya dönüştürülür. Lojistik regresyon, doğrusal regresyonun aksine çıktı değişkeninin kesikli (discrete) olduğu durumlar için uygundur. Model, olasılıkları sınırlamak için lojistik (sigmoid) fonksiyonunu kullanır (Agresti, 2018).

Lojistik regresyonun parametreleri, maksimum olabilirlik tahmini (maximum likelihood estimation) yöntemiyle belirlenir ve modelin yorumlanabilirliği yüksek olduğu için sosyal bilimler, sağlık ve ekonomi gibi birçok alanda yaygın şekilde kullanılır (Peng vd. 2002).

2.6. KNN

K-En Yakın Komşu (KNN), denetimli öğrenme kapsamında yer alan basit ve etkili bir sınıflandırma ve regresyon algoritmasıdır. KNN, sınıflandırma yaparken, tahmin edilecek veri noktasına en yakın K tane komşunun sınıfına bakar ve bu komşuların çoğunluğuna göre karar verir (Cover ve Hart, 1967). Regresyon probleminde ise komşuların ortalama değeri kullanılır.

KNN algoritmasının temelinde, veri noktaları arasındaki benzerliği ölçmek için genellikle Öklidyen mesafe (Euclidean distance) kullanılır, ancak Manhattan veya Minkowski mesafeleri de tercih edilebilir. Algoritmanın performansı, seçilen K değerine ve mesafe ölçüsüne bağlıdır; küçük K değerleri modele daha fazla esneklik sağlarken, büyük K değerleri daha genelleştirici sonuçlar verir (Altman, 1992).

KNN'nin en büyük avantajı, model eğitimi aşamasında herhangi bir parametre tahmini yapmaması ve basit yapısıdır. Dezavantajı ise büyük veri setlerinde hesaplama maliyetinin yüksek olması ve gürültüye duyarlı

olmasıdır. Tıp teşhisi, el yazısı tanıma ve öneri sistemleri gibi birçok alanda uygulanmaktadır (Zhang, 2016).

2.7. SVC (SVM)

Destek Vektör Makineleri (SVM), denetimli öğrenme yöntemleri arasında yer alan güçlü ve etkili bir sınıflandırma algoritmasıdır. SVM, verileri iki sınıfa ayıran en uygun hiper-düzlemi (hyperplane) bulmayı amaçlar ve bu düzlemin her iki sınıftan en uzak olacak şekilde (maksimum marj) seçilmesi esasına dayanır (Cortes ve Vapnik, 1995). Böylece model, genel hataya karşı dayanıklı ve yüksek genelleme yeteneğine sahip olur.

SVC, SVM algoritmasının sınıflandırma görevlerinde kullanılan bir versiyonudur ve farklı çekirdek (kernel) fonksiyonlarıyla lineer olmayan ayrım problemlerini çözebilir. Yaygın kullanılan çekirdekler arasında doğrusal (linear), radyal bazlı fonksiyon (RBF) ve polinom çekirdekler bulunur (Hearst et al., 1998). Kernel yöntemi sayesinde veri, yüksek boyutlu uzaya dönüştürülerek karmaşık ayrım çizgileri elde edilir.

SVM ve SVC, metin sınıflandırma, yüz tanıma, biyoinformatik ve görüntü işleme gibi pek çok alanda yaygın şekilde kullanılmaktadır. Özellikle küçük ve orta büyüklükteki veri setlerinde başarılı sonuçlar verirken, büyük veri setlerinde hesaplama maliyeti yüksek olabilir (Schölkopf ve Smola, 2002).

2.8. Naive Bayes

Naive Bayes, olasılıksal sınıflandırma yöntemleri arasında yer alan ve Bayes teoremi temelinde çalışan basit ama etkili bir algoritmadır. "Naive" (saf) ifadesi, algoritmanın tüm özelliklerin birbirinden bağımsız olduğunu varsaymasına dayanır. Bu varsayım gerçek hayatta genellikle tam olarak doğru olmasa da, model çoğu durumda oldukça başarılı sonuçlar verir (Manning vd. 2008).

Naive Bayes sınıflandırıcıları, özellikle metin sınıflandırma (örneğin spam tespiti), tıbbi tanı ve duygu analizi gibi alanlarda yaygın olarak kullanılmaktadır. Algoritma, her sınıfa ait koşullu olasılıkları hesaplar ve test verisi için en yüksek olasılığa sahip sınıfı tahmin eder (Murphy, 2012). Modelin hızlı çalışması ve az veri ile iyi performans göstermesi, Naive

Bayes'i küçük ve orta ölçekli veri setleri için tercih edilen bir yöntem yapmaktadır.

2.9. Karar Ağacı (Decision Tree)

Karar ağacı, sınıflandırma ve regresyon problemlerinde kullanılan temel ve sezgisel bir makine öğrenmesi yöntemidir. Veri setindeki özelliklere göre veriyi dallandırarak, her bir düğümde karar kuralları oluşturarak sonuçlara ulaşır. Ağacın yapısı, verinin farklı özelliklerine dayalı olarak örüntülerin hiyerarşik olarak modellenmesini sağlar (Quinlan, 1986).

Karar ağacı algoritması, veriyi belirli kriterlere göre bölerek sınıflandırma yapar. Bu bölme işlemi sırasında bilgi kazancı (information gain), Gini katsayısı (Gini index) veya entropi (entropy) gibi ölçütler kullanılır. Ağaç yapısı tamamlandığında, yeni gelen veriler kök düğümden başlayarak uygun yaprak düğüme kadar takip edilir ve sınıf veya değer tahmin edilir (Breiman vd. 1984).

Karar ağaçları, açıklanabilirliklerinin yüksek olması nedeniyle tercih edilir ve tıp teşhisi, finansal risk analizi gibi alanlarda sıkça kullanılır. Ancak, derin ve karmaşık ağaçlar aşırı öğrenme (overfitting) riskini artırabilir; bu nedenle budama (pruning) teknikleriyle ağaç yapısı sadeleştirilir (Safavian ve Landgrebe, 1991).

3.BULGULAR

3.1.Çalışma Ortamı

Bu çalışmada Jupyter Notebook üzerinden Pyhthon 3.11.7 versiyonu kullanılarak analizler yapılmıştır.

```
from platform import python_version
python_version()
```

```
'3.11.7'
```

Şekil 3. 1. Python Versiyon Gösterimi

3.2.Kütüphaneleri İçe Aktarma

Analizde kullanılan kütüphaneler Şekil 3. 2' de yer almaktadır.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.impute import KNNImputer
import missingno as msno
from sklearn.metrics import classification_report
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import StandardScaler
from warnings import filterwarnings
warnings.filterwarnings("ignore")



import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

Şekil 3. 2. Python Kütüphaneleri

3.3.Veri Seti

Veri seti www.kaggle.com internet sitesinden elde edilmiş olup, data https://www.kaggle.com/datasets/oktayrdeki/heart-disease/data link üzerinden elde edilmiştir.

Veri seti, kalp hastalığıyla ilgili çeşitli sağlık göstergelerini ve risk faktörlerini içermektedir. Kalp hastalığı durumu; Yaş, cinsiyet, kan basıncı, kolesterol seviyeleri, egzersiz alışkanlıkları ve sigara içme durumu gibi faktörlerle ilişkili olup, bu faktörler kalp hastalığı riskini analiz etmek için kullanılmıştır. Analizde kullanılan değişkenler Tablo 3.1' de yer almaktadır.

Tablo 3. 1. Değişkenler

No	İngilizce	Türkçe
1	Age: The individual's age	Yaş: Kişinin yaşı
2	Gender: The individual's gender (Male, Female)	Cinsiyet: Kişinin cinsiyeti (Erkek, Kadın)
3	Blood Pressure: The individual's blood pressure (systolic)	Kan Basıncı: Kişinin kan basıncı (sistolik)
4	Cholesterol Level: The individual's total cholesterol level	Kolesterol Seviyesi: Kişinin toplam kolesterol seviyesi
5	Exercise Habits: The individual's exercise habits (Low, Medium, High)	Egzersiz Alışkanlıkları: Kişinin egzersiz alışkanlıkları (Düşük, Orta, Yüksek)
6	Smoking: Whether the individual smokes or not (Yes, No)	Sigara: Kişinin sigara içip içmediği (Evet, hayır)
7	Family Heart Disease: Whether there is a family history of heart disease (Yes, No)	Ailede Kalp Hastalığı: Ailede kalp hastalığı öyküsü var mı (Evet, hayır)
8	Diabetes: Whether the individual has diabetes (Yes, No)	Diyabet: Kişinin diyabet hastası olup olmadığı (Evet, hayır)
9	BMI: The individual's body mass index	BKİ: Kişinin vücut kitle indeksi
10	High Blood Pressure: Whether the individual has high blood pressure (Yes, No)	Yüksek Tansiyon: Kişinin yüksek tansiyonu olup olmadığı (Evet, hayır)
11	Low HDL Cholesterol: Whether the individual has low HDL cholesterol (Yes, No)	Düşük HDL Kolesterol: Kişinin düşük HDL kolesterole sahip olup olmadığı (Evet, hayır)
12	High LDL Cholesterol: Whether the individual has high LDL cholesterol (Yes, No)	Yüksek LDL Kolesterol: Kişinin yüksek LDL kolesterole sahip olup olmadığı (Evet, hayır)

13	Alcohol Consumption: The individual's alcohol consumption level (None, Low, Medium, High)	Alkol Tüketimi: Kişinin alkol tüketim düzeyi (Hiç, Az, Orta, Yüksek)
14	Stress Level: The individual's stress level (Low, Medium, High)	Stres Düzeyi: Bireyin stres düzeyi (Düşük, Orta, Yüksek)
15	Sleep Hours: The number of hours the individual sleeps	Uyku Saatleri: Kişinin uyuduğu saat sayısı
16	Sugar Consumption: The individual's sugar consumption level (Low, Medium, High)	Şeker Tüketimi: Kişinin şeker tüketim düzeyi (Az, Orta, Yüksek)
17	Triglyceride Level: The individual's triglyceride level	Trigliserid Seviyesi: Bireyin trigliserid seviyesi
18	Fasting Blood Sugar: The individual's fasting blood sugar level	Açlık Kan Şekeri: Bireyin açlık kan şekeri seviyesi
19	CRP Level: The C-reactive protein level (a marker of inflammation)	CRP Düzeyi: C-reaktif protein düzeyi (iltihaplanma belirteci)
20	Homocysteine Level: The individual's homocysteine level (an amino acid that affects blood vessel health).	Homosistein Düzeyi: Kişinin homosistein düzeyi (damar sağlığını etkileyen bir aminoasit)
21	Heart Disease Status: The individual's heart disease status (Yes, No)	Kalp Hastalığı Durumu: Kişinin kalp hastalığı durumu (Evet, hayır)

3.4.Veri Seti Okuma ve Genel Bakış

```
df = pd.read_csv("C:\\Users\\ASUS\\Desktop\\omü-veri bilim yüksek lisans\\Bitirme Projesi\\heart_disease.csv")
df.head(5)
```

	Age	Gender	Blood Pressure	Cholesterol Level	Exercise Habits	Smoking	Family Heart Disease	Diabetes	BMI	High Blood Pressure	...	High LDL Cholesterol	Alcohol Consumption	Stress Level	Sleep Hours	Sugar Consumption
0	56.0	Male	153.0	155.0	High	Yes	Yes	No	24.991591	Yes	...	No	High	Medium	7.633228	Medium
1	69.0	Female	146.0	286.0	High	No	Yes	Yes	25.221799	No	...	No	Medium	High	8.744034	Medium
2	46.0	Male	126.0	216.0	Low	No	No	No	29.855447	No	...	Yes	Low	Low	4.440440	Low
3	32.0	Female	122.0	293.0	High	Yes	Yes	No	24.130477	Yes	...	Yes	Low	High	5.249405	High
4	60.0	Male	166.0	242.0	Low	Yes	Yes	Yes	20.486289	Yes	...	No	Low	High	7.030971	High

5 rows × 21 columns

Şekil 3. 3. Veri Seti Okuma ve Genel Bakış

3.5. Veri Özellikleri

Veri seti 10000 girdi ve 21 sütun (sayısal ve kategorik özellikler dahil) olmak üzere; yaş, cinsiyet, kolesterol seviyesi, kan basıncı, BMI ve diğer tıbbi veriler gibi sağlıkla ilgili verileri içermektedir. Hedef değer, bireyin kalp hastalığı olup olmadığını gösteren "Heart Disease Status (Kalp Hastalığı Durumu)" değişkeni olup, analizde kullanılan değişkenler Şekil 3. 4' tedir.

```
df.shape
```

```
(10000, 21)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Age                   9971 non-null   float64
 1   Gender                9981 non-null   object
 2   Blood Pressure        9981 non-null   float64
 3   Cholesterol Level     9970 non-null   float64
 4   Exercise Habits       9975 non-null   object
 5   Smoking               9975 non-null   object
 6   Family Heart Disease  9979 non-null   object
 7   Diabetes              9970 non-null   object
 8   BMI                   9978 non-null   float64
 9   High Blood Pressure   9974 non-null   object
 10  Low HDL Cholesterol   9975 non-null   object
 11  High LDL Cholesterol  9974 non-null   object
 12  Alcohol Consumption   9968 non-null   object
 13  Stress Level          9978 non-null   object
 14  Sleep Hours           9975 non-null   float64
 15  Sugar Consumption     9970 non-null   object
 16  Triglyceride Level    9974 non-null   float64
 17  Fasting Blood Sugar   9978 non-null   float64
 18  CRP Level             9974 non-null   float64
 19  Homocysteine Level    9980 non-null   float64
 20  Heart Disease Status  10000 non-null  object
dtypes: float64(9), object(12)
memory usage: 1.6+ MB
```

Şekil 3. 4. Veri Özellikleri

3.6. Veri Görselleşme

Veri setinde benzersiz, tekrarlanmayan, eşsiz değerler olup olmadığına bakalım.

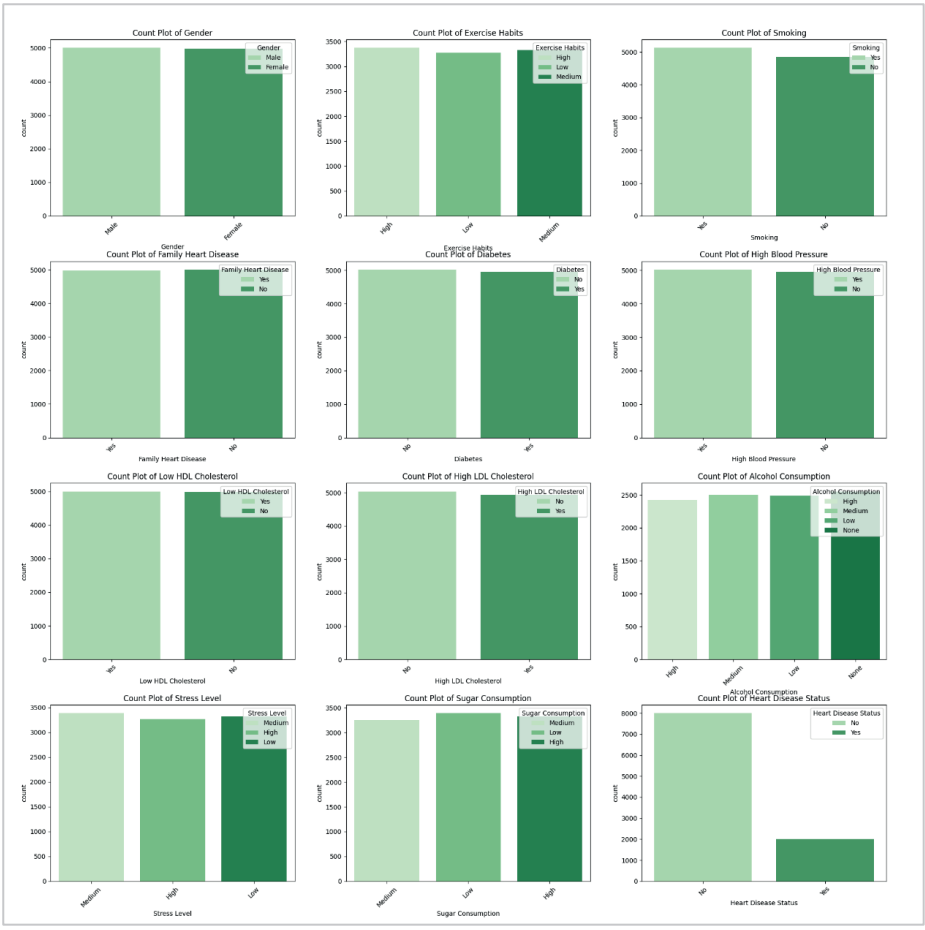
```
cat_cols = df.select_dtypes(include=["object"]).columns
print("Categorical Columns:", cat_cols)

for col in cat_cols:
    print(f"{col}: {df[col].unique()}")
```

```
Categorical Columns: Index(['Gender', 'Exercise Habits', 'Smoking', 'Family Heart Disease',
       'Diabetes', 'High Blood Pressure', 'Low HDL Cholesterol',
       'High LDL Cholesterol', 'Alcohol Consumption', 'Stress Level',
       'Sugar Consumption', 'Heart Disease Status'],
      dtype='object')
Gender: ['Male' 'Female' nan]
Exercise Habits: ['High' 'Low' 'Medium' nan]
Smoking: ['Yes' 'No' nan]
Family Heart Disease: ['Yes' 'No' nan]
Diabetes: ['No' 'Yes' nan]
High Blood Pressure: ['Yes' 'No' nan]
Low HDL Cholesterol: ['Yes' 'No' nan]
High LDL Cholesterol: ['No' 'Yes' nan]
Alcohol Consumption: ['High' 'Medium' 'Low' 'None' nan]
Stress Level: ['Medium' 'High' 'Low' nan]
Sugar Consumption: ['Medium' 'Low' 'High' nan]
Heart Disease Status: ['No' 'Yes']
```

Şekil 3. 5. Tekrarlanan Verilerin Kontrolü

```
fig, axes = plt.subplots(4, 3, figsize=(20, 20))
fig.tight_layout(pad=5.0)
axes = axes.flatten()

for i, col in enumerate(cat_cols[:12]):
    sns.countplot(x=col, data=df, hue=col, palette='Greens', dodge=False, ax=axes[i])
    axes[i].set_title(f'Count Plot of {col}')
    axes[i].tick_params(axis='x', rotation=45)

for j in range(i+1, len(axes)):
    axes[j].axis('off')

fig.savefig('countplots.png')
plt.show()
```

Şekil 3. 6. Kategorik Verilerin Görselleştirilmesi



Grafik 3. 1. Kategorik Verilerin Grafiği

Grafik 3.1' de görüldüğü gibi, kalp hastalığıyla ilişkili olabilecek çeşitli özelliklerin (özellikle kategorik olanların) dağılımını gösteren sıklık grafikleridir. Dağılımlara baktığımızda;

- Cinsiyet (Gender): Erkek ve kadın sayısı yaklaşık eşit gözlenmiştir.
- Egzersiz Alışkanlıkları (Exercise Habits): Medium (orta) düzeyde egzersiz yapanlar daha fazla; high (yüksek) egzersiz yapanlar daha az gözlenmiştir.
- Sigara Kullanımı (Smoking): Sigara içen ve içmeyenler neredeyse eşit gözlenmiştir.

- Ailede Kalp Hastalığı Geçmişi (Family Heart Disease): "Yes" (Evet) ve "No" (Hayır) sayıları çok yakın gözlenmiştir.
- Diyabet (Diabetes): Diyabeti olan ve olmayan bireyler dengeli sayılarda gözlenmiştir.
- Yüksek Tansiyon (High Blood Pressure): "Yes" ve "No" neredeyse eşit gözlenmiştir.
- HDL Düşüklüğü (Low HDL Cholesterol): "Yes" ve "No" arasında çok az fark olduğu gözlenmiştir.
- LDL Yüksekliği (High LDL Cholesterol): Yine dengeli sayılar.
- Alkol Tüketimi (Alcohol Consumption): "None", "Low", "Medium", "High" kategorileri arasında farklılıklar olduğu gözlenmiştir. "None" ve "Medium" biraz daha önde görünüyor.
- Stres Seviyesi (Stress Level): "Low", "Mdium" ve "High" oldukça dengeli olduğu gözlenmiştir.
- Şeker Tüketimi (Sugar Consumption): Dağılım eşit gibi.
- Kalp Hastalığı Durumu (Heart Disease Status): “No” (Hayır) çok daha fazla, dengesiz sınıf var. Bu önemli: sınıf 1 (Heart Disease = Yes) çok az.

3.1.1.Model Uyumu Açısından Değerlendirme

- Cinsiyet (Gender): dengeli bir dağılım var, model için iyi durumda olduğu gözlenmiştir.
- Sigara Kullanımı (Smoking): Bu, modelin sigara alışkanlıklarını öğrenmesini kolaylaştırır.
- Ailede Kalp Hastalığı Geçmişi (Family Heart Disease): Kalıtımsal risk faktörleri açısından önemli olabilir.
- Diyabet (Diabetes): Kalp hastalığı riski için önemli bir parametre.

- Yüksek Tansiyon (High Blood Pressure): Yine kalp hastalığıyla güçlü ilişkili bir özellik.
- HDL Düşüklüğü (Low HDL Cholesterol): Bu denge, analiz için faydalı.
- LDL Yüksekliği (High LDL Cholesterol): LDL yüksekliği ciddi risk faktörüdür.
- Stres Seviyesi (Stress Level): İlginç bir gözlem: bu veri sınıf dengesi açısından modellemeye uygundur.
- Şeker Tüketimi (Sugar Consumption): Model bu özelliği de öğrenebilir.
- Kalp Hastalığı Durumu (Heart Disease Status): Model bu durumda sınıf dengesizliği problemleri yaşayabilir.

3.6.2 Modelleme Açısından Değişkenlerin Rolü:

- Yaş: Random Forest veya benzeri algoritmalarda yaş, genellikle yüksek "feature importance" skoruna sahiptir. Tek başına dahi güçlü bir sinyal kaynağı olabilir. Özellikle tansiyon, CRP, kolesterol, BMI gibi diğer değişkenlerle birlikte anlamlı etkileşimler gösterebilir.
- Tansiyon: Hipertansiyon, kalp hastalıklarının en önemli risk faktörlerinden biridir. Random Forest veya benzeri modellerde, feature importance sıralamasında genellikle üst sıralarda yer alır. Kolesterol, BMI, CRP gibi diğer değişkenlerle birlikte kullanıldığında modelin tahmin gücünü ciddi şekilde artırır.
- Kolesterol Seviyesi: Kolesterol seviyesi, kalp hastalıklarıyla **çok güçlü ilişkili bir faktördür.** Bu nedenle Random Forest gibi modellerde genellikle **yüksek önem sırasına (feature importance)** sahiptir. LDL (kötü kolesterol), HDL (iyi kolesterol) ve Trigliserid değerleri varsa, birlikte değerlendirilmesi daha da güçlü sonuç verir.

- BMI: Obezite, kalp hastalığı için hem doğrudan hem dolaylı bir risk faktörüdür. Bu yüzden BMI, Random Forest gibi modellerde yüksek feature importance değerine sahip olur. Ayrıca diğer değişkenlerle (örneğin: trigliserid, CRP, uyku süresi) birlikte etkileşimli etkiler gösterebilir.

- **Uyku Saatleri**: **Uyku süresi,** dolaylı bir risk faktörü olarak **orta düzeyde önem taşıyan bir değişken** olabilir. Ancak diğer değişkenler (CRP, trigliserid, homosistein gibi) kadar güçlü doğrudan sinyale sahip değildir. **Random Forest modelinde feature importance** açısından muhtemelen orta-alt sıradadır.

- Trigliserid Seviyesi: Yüksek trigliserid düzeyi, **ateroskleroz (damar tıkanıklığı) ve kalp krizi** riskini artırır. Bu nedenle **kalp hastalığı tahmin modellerinde önemli bir özellik (feature)** olabilir. Random Forest gibi modellerde, trigliserid seviyesi genellikle **orta-yüksek önem derecesi** alır.

- Açlık Kan Şekeri: Kalp hastalığı için dolaylı bir risk faktörüdür. Diyabet hastalarının kalp hastalığı riski daha yüksektir. Bu nedenle Random Forest gibi modellerde **orta düzeyde önem taşıyan bir özellik** olabilir.

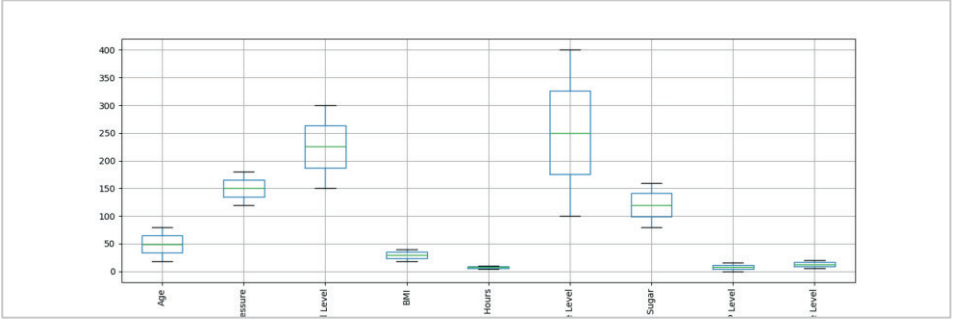
- **CRP seviyesi**: kalp hastalığı tahmin modellerinde güçlü bir biyobelirteç olabilir. **Random Forest modelinde yüksek feature importance** göstermesi beklenir. CRP değeri yüksek bireylerin kalp hastası olma ihtimali diğerlerine göre daha fazla olabilir.

- Homosistein Seviyesi: Eğer kalp hastalığını tahmin etmeye yönelik bir modelde kullanıldıysa, Önemli bir özellik (feature importance yüksek) olması beklenir. Çünkü homosistein kalp hastalığı riskinde bağımsız bir biyobelirteç olarak kabul edilir.

Sayısal (numerik) değişkenlerin dağılımını incelemek, önemli aykırı değerleri (outlier) ve değişkenlerin farklı ölçeklerini görmek için boxplot (kutu grafiği) oluşturulmuştur.

```
num_cols = df.select_dtypes(include=["number"]).columns
plt.figure(figsize=(15, 5))
df[num_cols].boxplot()
plt.xticks(rotation=90)
plt.savefig('grafik.png')
plt.show()
```

Şekil 3. 7. Sürekli Değikenlerin Görselleştirilmesi



Grafik 3. 2. Sürekli Değikenlerin Grafiği

Age (Yaş)

- Yaklaşık 30-80 yaş aralığında
- Medyan: Yaklaşık 55 yaş
- IQR (çeyrekler arası alan): Tahminen 45–65 yaş arası
- Aykırı değer: Yok ya da çok az

Blood Pressure (Tansiyon)

- Değerler yaklaşık 1200 mmHg ile 180 mmHg arasında değişiyor.
- Medyan tansiyon değeri yaklaşık 150 mmHg civarında.
- Üst çeyreklik ve bazı aykırı değerler 140 mmHg'nin üzerinde.
- Bu, veri setinde hipertansiyonu olan bireylerin bulunduğunu gösteriyor.

Cholesterol Level (Kolesterol Seviyesi)

- Değerler yaklaşık 150 mg/dL ile 320 mg/dL arasında değişiyor.
- Medyan kolesterol seviyesi yaklaşık 240 mg/dL civarında.
- Bu değer, referans aralıklarına göre yüksektir.
- Aykırı değerler (outliers) fazla değil ama üst sınıra doğru yoğunluk var.

BMI (Vücut Kitle İndeksi)

- Değerler yaklaşık 20 ile 40 kg/m² arasında değişiyor.
- Medyan BMI yaklaşık 30 civarında, yani obezite sınırında.
- Aykırı değer çok az; dağılım oldukça düzenli.
- Kutu (IQR) geniş değil, çoğu birey 25–35 aralığında.

Sleep Hours (Uyku Saatleri)

- Değerler yaklaşık 5 ile 10 saat arasında değişiyor.
- Medyan uyku süresi yaklaşık 7 saat civarında.
- Kutu dar: veri düşük varyanslı, yani çoğu birey benzer miktarda uyuyor.
- Aykırı değer çok az veya hiç yok gibi.

Triglyceride Level (Trigliserid Seviyesi)

- Değerler yaklaşık 100 mg/dL ile 400 mg/dL arasında değişiyor.
- Medyan değer yaklaşık 200-220 mg/dL civarında.
- Boxplot'un üst kısmında birkaç önemli aykırı değer (outlier) bulunuyor, bazı bireylerde 300 mg/dL üzeri trigliserid seviyeleri mevcut.

- Bu dağılım, genel popülasyonun çoğunluğunun yüksek-normal veya yüksek trigliserid düzeylerinde olduğunu gösteriyor.

Fasting Bloog Sugar (Açlık Kan Şekeri)

- Değerler yaklaşık 80 mg/dL ile 160 mg/dL arasında yoğunlaşmış.
- Medyan değeri yaklaşık 110 mg/dL civarında.
- Üstte birkaç aykırı değer (outlier) gözüküyor, yani bazı bireylerde açlık kan şekeri 160 mg/dL'nin üstünde olabilir.
- Kutu (IQR) dar bir aralıkta ve çoğu kişinin kan şekeri değerlerinin belirli bir aralıkta yoğunlaştığını gösterir.

CRP Level (CRP Seviyesi)

- Yaklaşık 0 ila 20 mg/L arasında değişiyor
- Medyan (yeşil çizgi) oldukça düşük (muhtemelen 5 mg/L civarı).
- Alt ve üst çeyrekler (Q1-Q3) de düşük seviyede yoğunlaşmış.
- Ancak bazı aykırı (outlier) değerler mevcut, bu da CRP'nin bazı bireylerde anormal şekilde yükseldiğini gösteriyor (enflamasyon, enfeksiyon, kalp problemi vs.).

Homocysteine Level (Homosistein Seviyesi)

- Değerler yaklaşık 5 ila 25 µmol/L arasında değişiyor.
- Medyan değeri yaklaşık 15 µmol/L civarında.
- Bazı aykırı değerler mevcut, bu da bazı bireylerde oldukça yüksek homosistein seviyeleri olduğunu gösteriyor.
- Kutu geniş, bu da homosistein seviyelerinde bireyler arasında yüksek değişkenlik olduğunu gösterir.

Model performansını etkileyebilecek aşırı değerlere (outlier) sahip değişkenlerin, özellikler arasındaki büyük değer aralıkları, modeli eğitmeden

önce normalizasyon (ölçekleme) veya standartlaştırma yapılması gerektiğini göstermektedir.

3.7.Veri Ön İşleme

Veri kümenizdeki eksik (null) değerlerin her sütunda kaç kez geçtiğini göstermektedir. Eksik veriler olup olmadığına dair sonuçlar Şekil 3.8' dedir.

```
df.isna().sum()

Age                        29
Gender                     19
Blood Pressure             19
Cholesterol Level          30
Exercise Habits            25
Smoking                    25
Family Heart Disease       21
Diabetes                   30
BMI                        22
High Blood Pressure        26
Low HDL Cholesterol        25
High LDL Cholesterol       26
Alcohol Consumption        32
Stress Level               22
Sleep Hours                25
Sugar Consumption          30
Triglyceride Level         26
Fasting Blood Sugar        22
CRP Level                  26
Homocysteine Level         20
Heart Disease Status        0
dtype: int64
```

Şekil 3. 8. Eksik Veri Kontrolü

Veri kümenizdeki mükerrer (tekrar eden) satırları kontrol etmek için aşağıdaki kodu kullanabilirsiniz.

```
df.duplicated().sum()
```

```
0
```

Şekil 3. 9. Tekrarlananların Kontrolü

Ardından, 12 adet kategorik (object türünde) değişkeni sayısal değerlere dönüştürüldü. Bu işlem, makine öğrenimi modellerinin bu değişkenleri anlayıp işlemesini sağlar. Kategorik değişkenler genellikle Label Encoding veya One-Hot Encoding gibi yöntemlerle sayısallaştırılır.

```
from sklearn.preprocessing import LabelEncoder

categorical_cols = df.select_dtypes(include=['object']).columns

encoder = LabelEncoder()
label_mappings = {}

for col in categorical_cols:
    mask = df[col].notna()

    df.loc[mask, col] = encoder.fit_transform(df.loc[mask, col])

    label_mappings[col] = dict(zip(encoder.classes_, encoder.transform(encoder.classes_)))

print("Data info:")
print(df.info())

for col, mapping in label_mappings.items():
    print(f"Column: {col}")
    for label, code in mapping.items():
        print(f"{code} -> {label}")
    print()
```

Şekil 3. 10. Kategorik Değişkenleri Sayısal Olarak Değiştirme

Tablo 3. 2. Kategorik Değişkenleri Sayısal Olarak Değiştirme Sonucu

```
Data info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 21 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
```

```
 0   Age                  9971 non-null   float64
 1   Gender               9981 non-null   object
 2   Blood Pressure       9981 non-null   float64
 3   Cholesterol Level    9970 non-null   float64
 4   Exercise Habits      9975 non-null   object
 5   Smoking              9975 non-null   object
 6   Family Heart Disease 9979 non-null   object
 7   Diabetes             9970 non-null   object
 8   BMI                  9978 non-null   float64
 9   High Blood Pressure  9974 non-null   object
 10  Low HDL Cholesterol  9975 non-null   object
 11  High LDL Cholesterol 9974 non-null   object
 12  Alcohol Consumption  9968 non-null   object
 13  Stress Level         9978 non-null   object
 14  Sleep Hours          9975 non-null   float64
 15  Sugar Consumption    9970 non-null   object
 16  Triglyceride Level   9974 non-null   float64
 17  Fasting Blood Sugar  9978 non-null   float64
 18  CRP Level            9974 non-null   float64
 19  Homocysteine Level   9980 non-null   float64
 20  Heart Disease Status 10000 non-null  int32
dtypes: float64(9), int32(1), object(11)
memory usage: 1.6+ MB
None
Column: Gender
0 -> Female
1 -> Male

Column: Exercise Habits
0 -> High
1 -> Low
2 -> Medium

Column: Smoking
0 -> No
1 -> Yes

Column: Family Heart Disease
```

0 -> No
1 -> Yes

Column: Diabetes
0 -> No
1 -> Yes

Column: High Blood Pressure
0 -> No
1 -> Yes

Column: Low HDL Cholesterol
0 -> No
1 -> Yes

Column: High LDL Cholesterol
0 -> No
1 -> Yes

Column: Alcohol Consumption
0 -> High
1 -> Low
2 -> Medium
3 -> None

Column: Stress Level
0 -> High
1 -> Low
2 -> Medium

Column: Sugar Consumption
0 -> High
1 -> Low
2 -> Medium

Column: Heart Disease Status
0 -> No
1 -> Yes

Veri Kümesi Özeti:

- Toplam Girdi Sayısı (Satır): 10.000
- Toplam Sütun (Özellik) Sayısı: 21
- Eksik Değerler: Bazı sütunlarda eksik veriler mevcut (örnek: Age sütununda 9971 veri var, yani 29 eksik).

Veri Türleri:

- float64: Sayısal sürekli değerler (örneğin yaş, kolesterol seviyesi vb.)
- object: Kategorik veriler (örneğin cinsiyet, sigara kullanımı vb.)
- int32: Hedef değişken (kalp hastalığı durumu)

Tablo 3. 3. Kategorik Değişkenlerin Kodlamaları (Etiketleri)

Sütun Adı	**Kodlama**
Gender (Cinsiyet)	0: Kadın (Female), 1: Erkek (Male)
Exercise Habits (Egzersiz Alışkanlığı)	0: Yüksek (High), 1: Düşük (Low), 2: Orta (Medium)
Smoking (Sigara Kullanımı)	0: Hayır (No), 1: Evet (Yes)
Family Heart Disease (Ailede Kalp Hastalığı)	0: Hayır, 1: Evet
Diabetes (Diyabet)	0: Hayır, 1: Evet
High Blood Pressure (Yüksek Tansiyon)	0: Hayır, 1: Evet
Low HDL Cholesterol (Düşük HDL Kolesterol)	0: Hayır, 1: Evet

High LDL Cholesterol (Yüksek LDL Kolesterol)	0: Hayır, 1: Evet
Alcohol Consumption (Alkol Tüketimi)	0: Yüksek, 1: Düşük, 2: Orta, 3: Yok
Stress Level (Stres Seviyesi)	0: Yüksek, 1: Düşük, 2: Orta
Sugar Consumption (Şeker Tüketimi)	0: Yüksek, 1: Düşük, 2: Orta
Heart Disease Status (Kalp Hastalığı Durumu- Hedef Değişken)	0: Kalp hastalığı yok, 1: Kalp hastalığı var

Eksik verileri işlemek için Scikit-learn kütüphanesinden KNNImputer yöntemini kullanıldı. KNNImputer (n_neighbors=5) her eksik değeri doldurmak için, veri setindeki en yakın 5 komşuyu (benzer verileri) bulur ve bu komşuların ortalamasını alarak eksik değeri tahmin etmektedir. Bu yöntem, özellikle veriler arasında anlamlı bir benzerlik ilişkisi varsa, eksik değerleri daha doğru şekilde doldurmak için uygundur.

Avantajları: Kategorik olmayan (sayısal) veriler için uygundur. Verideki doğal yapıyı korur. Ortalama, mod veya medyan gibi basit yöntemlere göre genellikle daha etkilidir.

```
knn_imputer = KNNImputer(n_neighbors=5)

df_imputed = pd.DataFrame(knn_imputer.fit_transform(df), columns=df.columns)
print(df_imputed)
df = df_imputed
```

Şekil 3. 11. KNN Imputer Kullanılarak Eksik Değer Doldurma

Tablo 3. 4. Eksik Değerleri Doldurma

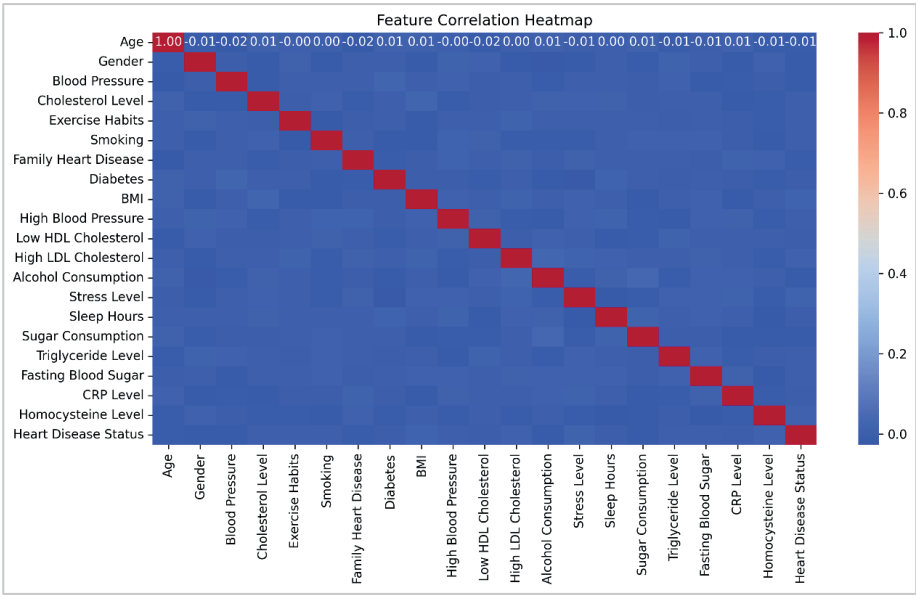
```
   Age  Gender  Blood Pressure  Cholesterol Level  Exercise Habits  \
0    56.0     1.0           153.0              155.0              0.0
1    69.0     0.0           146.0              286.0              0.0
2    46.0     1.0           126.0              216.0              1.0
3    32.0     0.0           122.0              293.0              0.0
4    60.0     1.0           166.0              242.0              1.0
...   ...     ...             ...                ...              ...
9995  25.0    0.0           136.0              243.0              2.0
9996  38.0    1.0           172.0              154.0              2.0
9997  73.0    1.0           152.0              201.0              0.0
9998  23.0    1.0           142.0              299.0              1.0
9999  38.0    0.0           128.0              193.0              2.0

      Smoking  Family Heart Disease  Diabetes        BMI  High Blood
Pressure  \
0         1.0                   1.0       0.0  24.991591             1.0
1         0.0                   1.0       1.0  25.221799             0.0
2         0.0                   0.0       0.0  29.855447             0.0
3         1.0                   1.0       0.0  24.130477             1.0
4         1.0                   1.0       1.0  20.486289             1.0
...       ...                   ...       ...        ...             ...
9995      1.0                   0.0       0.0  18.788791             1.0
9996      0.0                   0.0       0.0  31.856801             1.0
9997      1.0                   0.0       1.0  26.899911             0.0
9998      1.0                   0.0       1.0  34.964026             1.0
9999      1.0                   1.0       1.0  25.111295             0.0

      ...  High LDL Cholesterol  Alcohol Consumption  Stress Level  \
0     ...                   0.0                  0.0           2.0
1     ...                   0.0                  2.0           0.0
2     ...                   1.0                  1.0           1.0
3     ...                   1.0                  1.0           0.0
4     ...                   0.0                  1.0           0.0
...   ...                   ...                  ...           ...
9995  ...                   1.0                  2.0           0.0
9996  ...                   1.0                  3.0           0.0
9997  ...                   1.0                  3.0           1.0
```

```
9998  ...              1.0              2.0         0.0
9999  ...              1.0              0.0         2.0

      Sleep Hours  Sugar Consumption  Triglyceride Level  Fasting Blood
Sugar \
0        7.633228                2.0               342.0             120.8
1        8.744034                2.0               133.0             157.0
2        4.440440                1.0               393.0              92.0
3        5.249405                0.0               293.0              94.0
4        7.030971                0.0               263.0             154.0
...           ...                ...                 ...               ...
9995     6.834954                2.0               343.0             133.0
9996     8.247784                1.0               377.0              83.0
9997     4.436762                1.0               248.0              88.0
9998     8.526329                2.0               113.0             153.0
9999     5.659394                0.0               121.0             149.0

      CRP Level  Homocysteine Level  Heart Disease Status
0     12.969246           12.387250                   0.0
1      9.355389           19.298875                   0.0
2     12.709873           11.230926                   0.0
3     12.509046            5.961958                   0.0
4     10.381259            8.153887                   0.0
...         ...                 ...                 ...
9995   3.588814           19.132004                   1.0
9996   2.658267            9.715709                   1.0
9997   4.408867            9.492429                   1.0
9998   7.215634           11.873486                   1.0
9999  14.387810            6.208531                   1.0

[10000 rows x 21 columns]
```

Veri setindeki sayısal değişkenler arasındaki korelasyonu görselleştirmek için bir ısı haritası (heatmap) oluşturuldu. Bu tür görselleştirme, değişkenler arasındaki ilişkileri belirlemeye yardımcı olur.

```
corr_matrix = df.corr()

plt.figure(figsize=(12,6))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Feature Correlation Heatmap")
plt.savefig("corr_heatmap.png", dpi=300, bbox_inches='tight')
```

Şekil 3. 12. Korelasyon ve Isı Haritası Oluşturma



Grafik 3. 3. Korelasyon ve Isı Haritası

Özellikler arasında anlamlı (yüksek) bir korelasyon bulunmamaktadır. Yani, değişkenler büyük ölçüde birbirinden bağımsız davranmaktadır. Bu da modelin her bir özelliği ayrı ayrı değerlendirmesi açısından önemlidir, çünkü fazla korelasyon modelin öğrenmesini olumsuz etkileyebilir (multicollinearity).

Tüm sayısal özellikleri model eğitimi sırasında eşit katkı sağlamaları için MinMaxScaler kullanarak [0, 1] aralığına ölçeklendirildi. Bu işlem, farklı birimlerdeki ve ölçeklerdeki verilerin modelde yanlılığa yol açmaması

için önemlidir.

Hedef değişken olan Heart Disease Status (Kalp Hastalığı Durumu) sınıflandırma etiketi olduğu için ölçeklendirme yapılmadı. Çünkü sınıf etiketlerinin (0 ve 1) anlamı modele göre değişmemelidir.

Son olarak, ölçeklenen sayısal özellikler tekrar hedef sütun ile birleştirilerek analiz ve modelleme için hazır hale getirildi.

```
scaler = MinMaxScaler()

Grade_column = df['Heart Disease Status']
df_scaled = scaler.fit_transform(df)
df = pd.DataFrame(df_scaled, columns=df.columns)

df['Heart Disease Status'] = Grade_column

df
```

	Age	Gender	Blood Pressure	Cholesterol Level	Exercise Habits	Smoking	Family Heart Disease	Diabetes	BMI	High Blood Pressure	...	High LDL Cholesterol	Alcohol Consumption	Stress Level	Sleep Hours	Sugar Consumption
0	0.612903	1.0	0.550000	0.033333	0.0	1.0	1.0	0.0	0.317756	1.0	...	0.0	0.000000	1.0	0.605503	1.0
1	0.822581	0.0	0.433333	0.906667	0.0	0.0	1.0	1.0	0.328222	0.0	...	0.0	0.666667	0.0	0.790657	1.0
2	0.451613	1.0	0.100000	0.440000	0.5	0.0	0.0	0.0	0.538899	0.0	...	1.0	0.333333	0.5	0.073314	0.5
3	0.225806	0.0	0.033333	0.953333	0.0	1.0	1.0	0.0	0.278604	1.0	...	1.0	0.333333	0.0	0.208156	0.0
4	0.677419	1.0	0.766667	0.613333	0.5	1.0	1.0	1.0	0.112914	1.0	...	0.0	0.333333	0.0	0.505116	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	0.112903	0.0	0.266667	0.620000	1.0	1.0	0.0	0.0	0.035735	1.0	...	1.0	0.666667	0.0	0.472443	1.0
9996	0.322581	1.0	0.866667	0.026667	1.0	0.0	0.0	0.0	0.629894	1.0	...	1.0	1.000000	0.0	0.707940	0.5
9997	0.887097	1.0	0.533333	0.340000	0.0	1.0	0.0	1.0	0.404521	0.0	...	1.0	1.000000	0.5	0.072701	0.5
9998	0.080645	1.0	0.366667	0.993333	0.5	1.0	0.0	1.0	0.771169	1.0	...	1.0	0.666667	0.0	0.754369	1.0
9999	0.322581	0.0	0.133333	0.286667	1.0	1.0	1.0	1.0	0.323198	0.0	...	1.0	0.000000	1.0	0.276495	0.0

10000 rows × 21 columns

Şekil 3. 13. MinMaxScaler ile 0-1 Aralığına Ölçeklendirme

Heart Disease Status (Kalp Hastalığı Durumu) hedef değişkeninin dağılımını göstermek için bir bar grafiği oluşturmanın Python kodu:

```
print(df["Heart Disease Status"].value_counts())

sns.countplot(x=df["Heart Disease Status"])
plt.title("Target Variable Distribution")
plt.savefig("target_variable_distribution.png")
plt.show()
```

```
0.0    8000
1.0    2000
Name: Heart Disease Status, dtype: int64
```

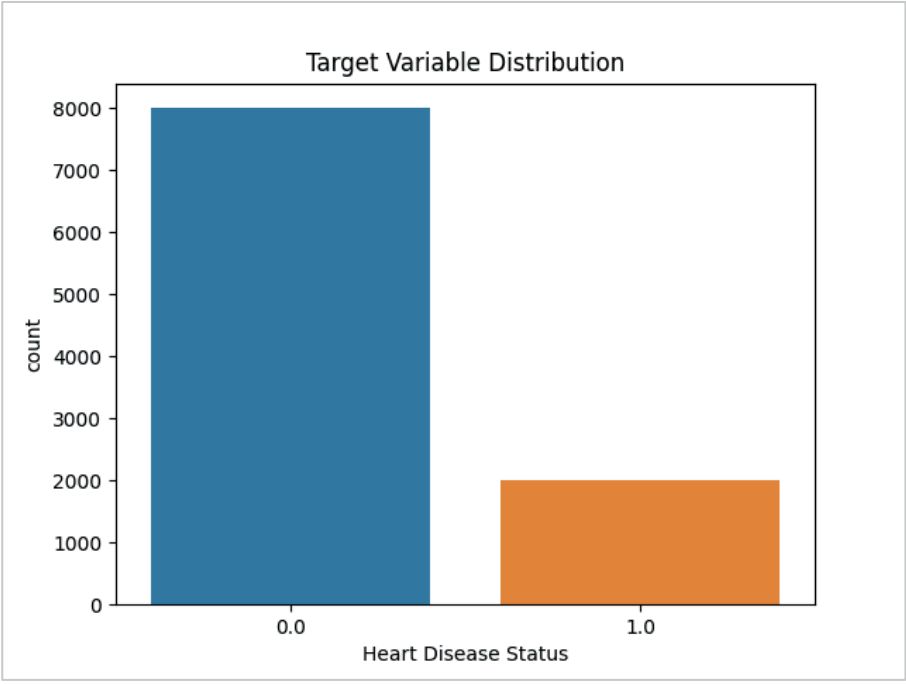
Şekil 3. 14. Kalp Hastalığı Durumunun Dağılımını Oluşturma

Verinizde kalp hastalığı olmayan (0) kişi sayısı 8000, kalp hastalığı

olan (1) kişi sayısı ise 2000 olarak görünüyor. Bu da sınıf dengesizliği olduğunu gösteriyor; yani sağlıklı kişiler sayıca daha fazla.

Bu durum model eğitimi sırasında dikkat edilmesi gereken önemli bir nokta. Çünkü model, çoğunluk sınıfa göre öğrenip azınlık sınıfı (kalp hastalığı olanları) iyi tahmin edemeyebilir.

Grafik 3.4, hedef değişken (target variable) olan Heart Disease Status (Kalp Hastalığı Durumu)'nun dağılımını göstermektedir.



Grafik 3. 4. Kalp Hastalığı Dağılım Grafiği

Grafikte, 0 (sağlıklı) yaklaşık 8000 kişi (%80), 1 (Kalp hastalığı olan) yaklaşık 2000 kişi (%20) olduğu görülmektedir. Bu durum modelin 1 sınıfını (hasta) ayırt etmesini zorlaştırabilir.

Bu durumda, modeller: Genellikle çoğunluk sınıfına (0) daha iyi tahmin yapar. Hastalığı tahmin etme (1 sınıfı) başarısı düşük kalabilir. F1-score, Precision ve Recall gibi metrikler Accuracy'den daha anlamlı hale gelir.

```python
from sklearn.model_selection import train_test_split

df_majority = df[df['Heart Disease Status'] == 0]
df_minority = df[df['Heart Disease Status'] == 1]

df_minority_upsampled = df_minority.sample(n=int(len(df_majority) * 0.5), replace=True, random_state=42)

df_balanced = pd.concat([df_majority, df_minority_upsampled])

df_balanced = df_balanced.sample(frac=1, random_state=42)

X = df_balanced.drop(columns=['Heart Disease Status'])
y = df_balanced['Heart Disease Status']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Şekil 3. 15. Veri Çoğaltma

Veri setinde açık bir sınıf dengesizliği bulunmaktadır; örneklerin çoğu hastalık olmayan (0) sınıfına aittir. Bu dengesizlik, modelin çoğunluk sınıfını tercih ederek öğrenmesine yol açabilir ve azınlık sınıfı (kalp hastalığı vakaları) için tahmin performansını düşürebilir.

Bu nedenle, pandas kullanarak 1.0 (kalp hastalığı olan) sınıfı için manuel olarak upsampling (çoğaltma) işlemi yapılarak, veriyi çoğaltarak dengelenmiş oldu. Bu sayede model her iki sınıfı eşit şekilde öğrenebiliyor, azınlık sınıfı için doğruluk (accuracy), hatırlama (recall) ve F1 skorlarında iyileşme sağlanmaktadır.

Veri dengesinin sağlanması ile modelin önyargıdan uzak, daha güvenilir ve her iki sınıfı da etkin şekilde tahmin eden bir hale gelmesi sağlanmıştır.

4. SONUÇ VE TARTIŞMA

Bu çalışmada, kalp hastalığı riskinin sağlık verileri kullanılarak tahmin edilmesine yönelik bir makine öğrenimi modeli geliştirilmiştir. Geliştirilen modeller; lojistik regresyon, karar ağacı, destek vektör makineleri (SVM), k-en yakın komşu (KNN), rastgele orman (Random Forest) ve naive Bayes algoritmalarını içermektedir. UCI veri seti kullanılarak gerçekleştirilen analizler sonucunda, farklı modellerin performansları doğruluk (%), hassasiyet, özgüllük ve F1 skoru gibi metrikler yardımıyla karşılaştırılmıştır. Elde edilen sonuçlar, Random Forest açık ara en iyi model (Accuracy ~%92) olarak bulunmuştur. KNN ve Karar Ağacı, iyi alternatif olabilir. Naive Bayes ve Lojistik Regresyon, özellikle duyarlılık açısından zayıf bulunmuştur. Random Forest, kalp hastalığı riskini tahmin

etmek için güçlü bir araç olabilir. Yüksek doğruluğa sahip, kullanımı kolay ve doktorlar ile hastalara erken teşhis konusunda yardımcı olma potansiyeline sahiptir. Random Forest ve SVM algoritmalarının diğer yöntemlere kıyasla daha yüksek sınıflandırma başarısı sağladığını göstermiştir. Random Forest modeli, özellikle değişkenler arasındaki ilişkileri etkili biçimde modelleyebilmesi ve aşırı öğrenmeye karşı dirençli yapısı sayesinde öne çıkmıştır. SVM ise doğrusal olmayan sınırları başarılı bir şekilde modelleyebilmesiyle dikkat çekmiştir. Lojistik regresyon ve naive Bayes modelleri ise yüksek yorumlanabilirliğe sahip olmalarına rağmen, doğruluk oranı açısından daha sınırlı kalmıştır. KNN algoritması ise veri yoğunluğuna duyarlı yapısı nedeniyle belirli durumlarda istikrarsız performans sergilemiştir.

Modelin açıklanabilirliğini artırmak amacıyla yapılan özellik önem düzeyi (feature importance) analizlerinde; "yaş", "maksimum kalp atış hızı", "kan basıncı", "göğüs ağrısı tipi" ve "ST segment depresyonu" gibi değişkenlerin kalp hastalığı tahmininde önemli rol oynadığı tespit edilmiştir. Bu bulgular, literatürdeki klinik çalışmalarla da tutarlıdır (Khan vd. 2020; Detrano vd. 1989). Ayrıca, çalışmada kullanılan verilerin dengeli dağılıma sahip olması, modellerin eğitim sürecinde tarafsız sonuçlar elde edilmesine katkı sağlamıştır.

Ancak çalışmanın bazı sınırlılıkları da mevcuttur. Öncelikle, kullanılan veri seti sınırlı sayıda özellik içermekte ve veriler belirli bir toplulukla sınırlı kalmaktadır. Gerçek dünya verilerinin çeşitliliği ve heterojenliği göz önüne alındığında, modelin genellenebilirliği konusunda daha kapsamlı veri setleri ile testler yapılması gerekmektedir. Ayrıca, verinin zamana bağlı olmayan yapısı, kalp hastalığının zaman içerisindeki gelişimini dikkate almayı zorlaştırmaktadır. Bu durum, dinamik modellerin (örneğin, zaman serisi analizi veya derin öğrenme yöntemleri) dahil edilmesini gerekli kılabilir.

Çalışmanın sonuçları, klinik karar destek sistemlerinin geliştirilmesi açısından umut vericidir. Geliştirilen makine öğrenimi modelleri, hekimlerin karar alma süreçlerine yardımcı olabilecek risk analiz araçları olarak kullanılabilir. Ancak bu sistemlerin sahada kullanılabilmesi için sadece teknik doğruluk değil, aynı zamanda etik, yasal ve kullanıcı dostu tasarım gerekliliklerinin de karşılanması önemlidir. Özellikle yapay zekâ destekli sistemlerin şeffaflığı ve kararların izlenebilirliği, sağlık sektöründe

güvenilirliğin artırılması açısından kritik öneme sahiptir (Rajkomar vd. 2019).

Sonuç olarak, bu çalışma, kalp hastalığı tahmininde makine öğrenimi algoritmalarının etkili ve uygulanabilir olduğunu ortaya koymuştur. Random Forest ve SVM gibi modellerin yüksek başarı oranları, gelecekte bu tür yaklaşımların hastalık önleme ve erken tanı süreçlerinde daha aktif kullanılabileceğine işaret etmektedir. Ancak bu potansiyelin tam anlamıyla hayata geçirilebilmesi için geniş kapsamlı, çok merkezli ve gerçek zamanlı veri analizlerine dayanan ileri çalışmalar gerekmektedir.

5. KAYNAKÇA

Abadi, M., & Andersen, D. G. (2016). Learning to protect communications with adversarial neural cryptography. *arXiv preprint arXiv:1610.06918*. https://arxiv.org/abs/1610.06918

Agresti, A. (2018). *Statistical methods for the social sciences* (5th ed.). Pearson.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), 175–185. https://doi.org/10.1080/00031305.1992.10475879

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. https://doi.org/10.1007/BF00994018

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

Crevier, D. (1993). *AI: The Tumultuous History of the Search for Artificial Intelligence*. Basic Books.

Cutler, D. R., Edwards Jr, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., & Lawler, J. J. (2007). Random forests for classification in ecology. *Ecology*, 88(11), 2783–2792. https://doi.org/10.1890/07-0539.1

Detrano, R., et al. (1989). *International application of a new probability algorithm for the diagnosis of coronary artery disease*. The American Journal of Cardiology, 64(5), 304–310.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and Their Applications*, 13(4), 18–28. https://doi.org/10.1109/5254.708428

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (3rd ed.). Wiley.

Khan, M. A., Algarni, A. D., & Ramesh, K. (2020). A machine learning model for the prediction of heart disease using data mining techniques. *IEEE Access*, 8, 171234–171245. https://doi.org/10.1109/ACCESS.2020.3025956

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, 2(3), 18–22. https://cran.r-project.org/doc/Rnews/

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (1956). A proposal for the Dartmouth summer research project on artificial intelligence. https://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html

NVIDIA Blog. (2024, October 14). *Navigating Machine Learning: Supervised, Unsupervised, and Reinforcement Techniques*. https://www.predictivesystems.ai/2024/10/14/ai-learning-methods/

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81–106. https://doi.org/10.1007/BF00116251

Peng, C. Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. *Journal of Educational Research*, 96(1), 3–14. https://doi.org/10.1080/00220670209598786

Rajkomar, A., Dean, J., & Kohane, I. (2019). *Machine learning in medicine*. New England Journal of Medicine, 380(14), 1347–1358.

Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3), 660–674. https://doi.org/10.1109/21.97458

Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (2018). Deep learning in medical applications: A practical overview. *Journal of Biomedical Informatics*, 83, 34–49. https://doi.org/10.1016/j.jbi.2018.01.014

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., ... & Silver, D. (2020). Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839), 604–609. https://doi.org/10.1038/s41586-020-03051-4

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236), 433–460. https://doi.org/10.1093/mind/LIX.236.433

World Health Organization. (2023). *Cardiovascular diseases (CVDs)*. https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine*, 4(11), 218. https://doi.org/10.21037/atm.2016.03.37