

## Kar Erime Optimizasyonu Algoritması ile Çok Katmanlı Yapay Sinir Ağının Eğitimi

Feyza Nur ÖZDEMİR<sup>1</sup> ORCID 0000-0001-7803-7725  
Ahmet ÖZKİŞ<sup>\*2</sup> ORCID 0000-0002-1899-5494

<sup>1</sup>Necmettin Erbakan Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Konya

<sup>2</sup>Necmettin Erbakan Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Konya

Geliş tarihi: 25.01.2024 Kabul tarihi: 27.06.2024

Atıf şekli/ How to cite: ÖZDEMİR, F.N., ÖZKİŞ, A., (2024). Kar Erime Optimizasyonu Algoritması ile Çok Katmanlı Yapay Sinir Ağının Eğitimi. Çukurova Üniversitesi, Mühendislik Fakültesi Dergisi, 39(2), 391-406.

### Öz

Kar erime optimizasyonu (snow ablation optimizer, SAO) algoritması, karın erimesinden ilham alınarak 2023 yılında önerilen yeni bir metasezgisel algoritmadır. Bu çalışmada, yapay sinir ağının ağırlıklarının güncellenmesi amacıyla SAO algoritması ile hibrit bir model geliştirilmiştir. Geliştirilen hibrit model aggregation, balance, liver, pathbased ve wine adlı beş farklı veri seti üzerinde gri kurt, sürüngen arama, guguk kuşu ve sinüs kosinüs algoritmaları ile oluşturulan hibrit modeller ile karşılaştırılmıştır. Modellerin başarısını ölçmek için duyarlılık, özgüllük, kesinlik ve f1-puanı adı verilen dört farklı metrik kullanılmıştır. Modellerin her veri seti için başarı sıralaması ve tüm veri setleri için ortalama başarı sıralaması verilmiştir. Sonuçlar incelendiğinde, SAO modelinin tüm metrikler için wine veri setinde 2., diğer tüm veri setlerinde 1. olduğu görülmektedir. Ortalama başarı sırası açısından ise SAO modeli tüm metriklerde 1.2 değeri ile en iyi sonucu elde etmiştir. Ayrıca hibrit modellerin, eğitim aşamasındaki ortalama karesel hata değerlerinin yakınsama grafikleri çizdirilmiş ve SAO hibrit modelinin wine hariç diğer tüm veri setlerinde karşılaştırılan modellerden daha hızlı bir yakınsama performansına sahip olduğu gözlenmiştir. Son olarak popülasyondaki parçacık sayısının hibrit SAO modelinin başarısına etkisi analiz edilmiş ve birey sayısının 100 olması durumunda başarının arttığı gözlenmiştir.

**Anahtar Kelimeler:** Kar erime optimizasyonu algoritması, Yapay sinir ağları, Makine öğrenmesi, Danışmanlı öğrenme, Sınıflandırma

### Training of Multilayer Artificial Neural Network with Snow Ablation Optimizer Algorithm

#### Abstract

The snow ablation optimizer (SAO) is a new metaheuristic algorithm proposed in 2023, inspired by snow melting. In this study, a hybrid model was developed with the SAO algorithm to update the artificial neural

\*Sorumlu yazar (Corresponding Author): Ahmet ÖZKİŞ, aozkis@erbakan.edu.tr

network weights. The developed hybrid model was compared with hybrid models created with gray wolf, reptile search, cuckoo and sine cosine algorithms on five different data sets named aggregation, balance, liver, pathbased and wine. Four different metrics called sensitivity, specificity, precision and f1-score were used to measure the success of the models. The success ranking of the models for each data set and the average success ranking for all data sets are given. When the results are examined, it is seen that the SAO model ranks 2nd in the wine data set and 1st in all other data sets for all metrics. Regarding average success rank, the SAO model achieved the best result with a value of 1.2 in all metrics. In addition, convergence graphs of the mean square error values of the hybrid models in the training phase were drawn and it was observed that the SAO hybrid model had a faster convergence performance than the compared models in all other data sets except Wine. Finally, the effect of the number of particles in the population on the success of the hybrid SAO model was analyzed and it was observed that the success increased when the number of individuals was 100.

**Keywords:** Snow ablation optimizer algorithm, Artificial neural networks, Machine learning, Supervised learning, Classification

## 1. GİRİŞ

İnsanlık var olduğundan beri çok çeşitli problemlerle karşı karşıya kalmıştır. Yakın geçmişimize kadar bu problemlerin çözümü için sadece kendi akıl ve zekasını kullanabilen insanoğlu, dijitalleşme ile adeta çağ atlayarak çoğu problemin çözümünde bilgisayarları kullanmaya başlamıştır. Teknolojinin ilerlemesiyle beraber depolanabilen veri miktarının da çok hızlı bir şekilde artması, Turing [1] başta olmak üzere bilim insanlarına şu soruyu sordurmuştur: “İnsanın yapabildiği bazı görevler makinelerle öğretilbilir mi?” Bu soru, günümüzdeki yapay zekâ kavramına giden yolu açmıştır. Yapay zekâ, insan ve hayvan gibi biyolojik canlılardaki karar verme, tahmin etme ve çözüm bulma kabiliyetlerinin makinelerle öğretilmesidir [2]. Bu amacı gerçekleştirmek için bilim insanları çeşitli çalışmalar yapmıştır. Rosenblatt [3] tarafından geliştirilen basit algılayıcı modeli yapay zekâ alanının öncü çalışmalarından biri olmuştur. Rumelhart ve arkadaşları [4] ise perceptron modelinin doğrusal olmayan problemleri çözmesini sağlayarak yapay sinir ağları (YSA) algoritmasının geliştirilmesine katkı sunmuştur. YSA algoritması günümüzde görüntü işleme [5-7], zaman serisi tahmini [8-10], regresyon analizi [11-14], sınıflandırma [15-17] ve kümeleme [18-20] gibi pek çok gerçek dünya probleminin çözümünde başarıyla uygulanmaktadır.

YSA mimarisi, nöron denilen yapılardan oluşur. Nöronlar ile girdi, ara ve çıktı katmanları

oluşturulur. YSA mimarisi üzerinde hesap edilen hata değerlerinin girdi katmanından çıktı katmanına doğru aktarılmasına *ileri yayılım* denir. İleri yayılım sonucunda hesaplanan hatanın azaltılması için nöronlardaki *ağırlık* adı verilen sayısal değerlerin güncellenmesi işlemine ise *geri yayılım* denilmektedir. YSA modelinde, geri yayılım esasında ağırlıkların güncellenmesi için genellikle türev tabanlı teknikler kullanılmaktadır [21]. Ancak türev tabanlı teknikler zorlayıcı problemlerle karşılaştıklarında ağırlık değerlerinin başarıyla güncellenmesi mümkün olamamakta ve ağırlık değerleri yerel minimum noktalara takılabilmektedir [2]. Bu sorunun üstesinden gelmek için, araştırmacılar pek çok çalışmada [21-24] ağırlık geri yayılımı aşamasında metasezgisel algoritmalar kullanmışlardır.

Metasezgisel yaklaşımlar, canlıların besin arama ve hayatta kalma davranışlarından ilham alınarak bilim insanları tarafından geliştirilen doğa esinli algoritmalarlardır. Bu algoritmalar hızlı, esnek ve kolay uygulanabilir yapılarından dolayı sayısal optimizasyon [25], kapasitesiz tesis yerleştirme [26, 27], rüzgâr türbini yerleştirme [28], knapsack [29] ve YSA'nın eğitilmesi [30] gibi birçok gerçek dünya probleminin çözümünde başarı ile uygulanabilmektedir. Günümüze kadar çok sayıda metasezgisel algoritma geliştirilmiş olsa da No Free Lunch (NFL) [31] teoremindeki, hiçbir algoritma tüm problem türlerinde en başarılı sonuca ulaşmayı garanti edemez düşüncesinden hareketle halen yeni algoritmalar önerilmektedir. Deng ve arkadaşları

[25] tarafından önerilen kar erime optimizasyonu (snow ablation optimizer, SAO) algoritması da NFL teoreminden hareketle önerilen yeni bir metasezgisel algoritmadır. Bu çalışmada, YSA'nın geri yayılımında ağırlıkların yerel minimuma takılma sorununu ortadan kaldırmak için SAO algoritması tabanlı hibrit bir yöntem geliştirilmiştir. Yöntemin performans testi için aşağıdaki adımlar uygulanmıştır:

- Geliştirilen hibrit model literatürde iyi bilinen aggregation, balance, liver, pathbased ve wine sınıflandırma veri setleri üzerinde çalıştırılmıştır.
- Eğitim ve test aşamalarında elde edilen sonuçlar için sensitivity, specificity, precision ve f1-score metrikleri hesaplanmıştır.
- Geliştirilen modelin metrik sonuçları, guguk kuşu arama (cuckoo search, CS) algoritması, gri kurt optimizasyonu (grey wolf optimizer, GWO), sürüngen arama algoritması (reptile search algorithm, RSA) ve sinüs kosinüs algoritması (sine cosine algorithm, SCA) ile oluşturulan hibrit modellerin sonuçlarıyla karşılaştırılmıştır.
- Modellerin, YSA'nın eğitimi esnasında her iterasyondaki ortalama karesel hata (mean squared error, MSE) değerleri üzerinden yakınsama grafikleri verilmiştir.

Elde edilen sonuçlar SAO tabanlı hibrit modelin karşılaştırılan diğer modellerden çok daha üstün bir performans elde ettiğini göstermiştir.

Çalışmanın geri kalanı şu şekilde organize edilmiştir: Bölüm 1.1'de literatür özeti sunulmuştur. Bölüm 2'de materyal ve metot hakkında bilgilere yer verilmiştir. Bölüm 3'te yapılan deneysel çalışmalar ve sonuçları verilmiştir. Bölüm 4'te ise tartışma ve sonuç kısmına yer verilmiştir.

### 1.1. Literatür Özeti

Gerçek dünyada, YSA ve metasezgisel algoritmaların hibrit kullanımıyla ilgili pek çok çalışma vardır. Singh ve arkadaşları [32] yeşil hindistan cevizi kabuklarının antioksidan aktivitesini maksimuma taşıyabilmek için YSA ve

parçacık sürü optimizasyonu (particle swarm optimization, PSO) algoritmasını kullanmıştır. Bendine ve arkadaşları [33] düzgün olmayan fonksiyonel derecelendirilmiş piezoelektrik ışının enerji hasadının geliştirilmesinde YSA ve Lichtenberg algoritmasını kullanmışlardır. Yang ve arkadaşları [34], maya fermantasyonundaki çok aşamalı gözenekli hidrojellerin hazırlama koşullarını optimize etmek için YSA ve genetik algoritma (genetic algorithm, GA)'dan faydalanmıştır. Wang ve arkadaşları [35], katı oksit yakıt hücresinde iç sıcaklık farkını azaltmak amacıyla YSA ve GA tabanlı bir model önermişlerdir. Cinar ve arkadaşları [36], beş şehrin 38 yıla ait saatlik rüzgâr verileri ile Hindistan'daki saatlik rüzgâr hızını tahmin etmek için YSA ve GWO algoritmasını birlikte kullanmıştır. Bernard ve arkadaşları [37] ise öğrencilerin öğrenme stillerini belirlemek için YSA ve karınca kolonisi optimizasyonu (ant colony optimization, ACO) algoritmasını kullanarak hibrit bir model geliştirmişlerdir. Zhang ve arkadaşları [38] YSA ile ACO algoritmasını birlikte kullanarak açık ocak madencilik projelerinin sermaye maliyetini tahmin etmek için bir model önermişlerdir.

Yukarıdaki çalışmaların yanı sıra sınıflandırma veri setleri üzerinde test edilmiş pek çok çalışma da vardır. Mirjalili [39], GWO algoritmasını 5 sınıflandırma ve 3 benchmark veri seti üzerinde test etmiştir. Gülcü [40], YSA'nın ağırlıklarını yusufçuk algoritması (dragonfly algorithm, DA) ile optimize etmiştir. Önerdiği yöntemi 8 farklı sınıflandırma problemi ve bir inşaat mühendisliği probleminde uygulamıştır. Turkoglu ve arkadaşları [41], yapay alg algoritması (artificial algae algorithm, AAA) ile YSA'yı eğitmiş ve önerdikleri yöntemi 10 farklı sınıflandırma veri seti üzerinde test etmişlerdir.

## 2. MATERYAL VE METOT

### 2.1. Veri Setleri

Çalışmada, literatürde sıklıkla kullanılan [42] aggregation, balance, liver, pathbased ve wine veri setleri kullanılmıştır. Veri setlerinin %75'i eğitim, %25'i test aşaması için ayrılmıştır. Veri setleriyle ilgili diğer bilgiler Çizelge 1'de sunulmuştur.

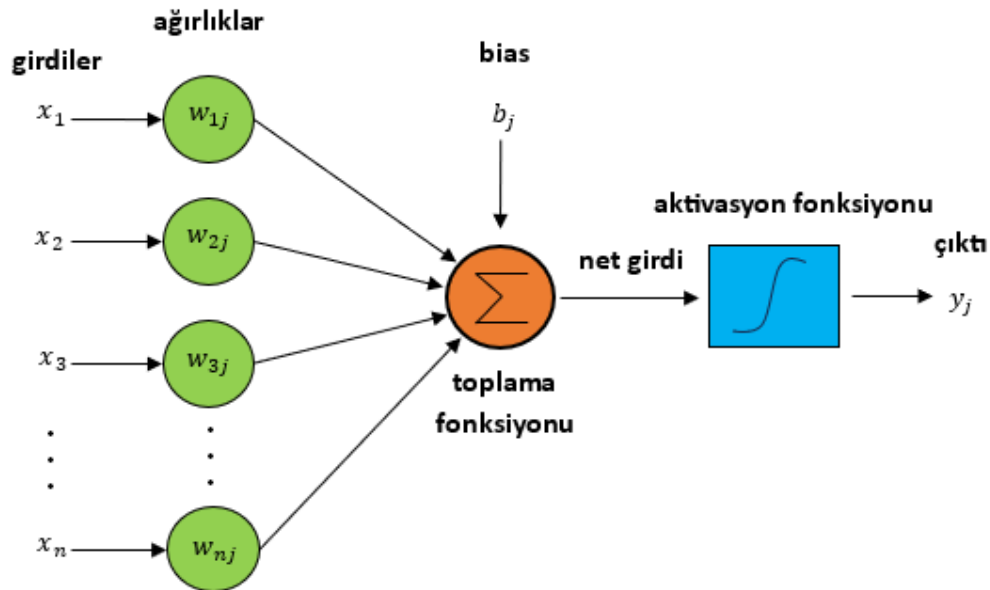
## 2.2. Yapay Sinir Ağları (YSA)

Yapay sinir ağları, insan beynindeki sinir hücrelerinin öğrenme yöntemini taklit eden bir algoritmadır. YSA, mevcut verilerden öğrenebilme, öğrendiği bilgilerden genellemeler yapabilmeye, çok fazla değişkenle çalışabilme gibi avantajlara sahiptir. Doğrusal ve doğrusal olmayan problemleri çözebilmektedir. Performansını kesin olarak neyin

etkilediği hakkında tam olarak bilgi sahibi olunamasa da birçok çalışmada eğitim algoritması, veri kümesinin düzenlenmesi, kestirim dönemi uzunluğu gibi faktörlerin performansını etkileyebileceği belirtilmiştir [43]. YSA'nın en küçük birimi yapay sinir hücresi olarak adlandırılır. Şekil 1'de bir yapay sinir hücresinin; girdi, çıktı, toplama fonksiyonu, aktivasyon fonksiyonu, ağırlık ve bias değerleri görülmektedir [2].

**Çizelge 1.** Veri setleri ve özellikleri

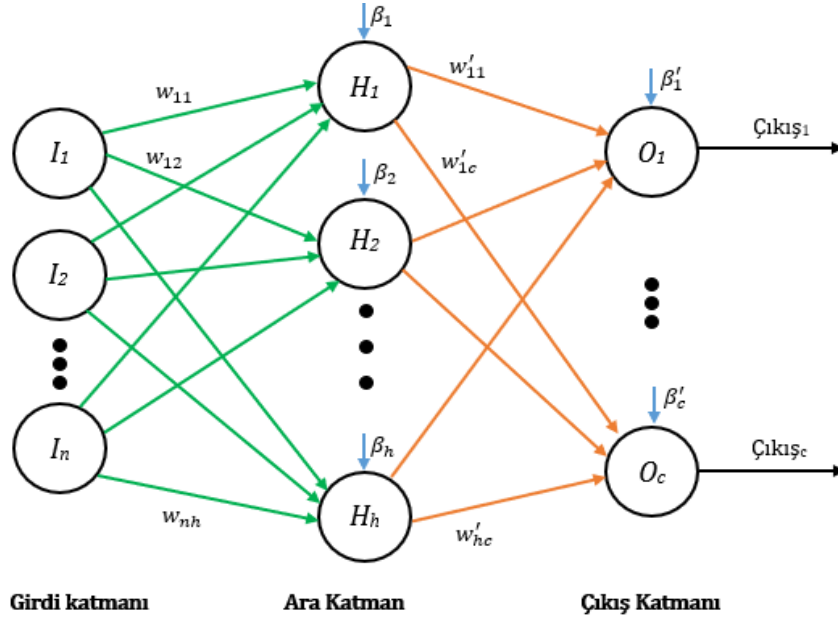
	Örnek sayısı	Eğitim örneği sayısı	Test örneği sayısı	Öznitelik sayısı	Sınıf sayısı
aggregation	788	594	194	2	7
balance	625	469	156	4	3
liver	345	259	86	6	3
pathbased	400	226	74	2	3
wine	178	133	45	13	3



**Şekil 1.** Yapay sinir hücresi

YSA'nın yapısı, tek katmanlı algılayıcı (TKA) ve çok katmanlı algılayıcı (ÇKA) olarak ikiye ayrılmaktadır. YSA hakkındaki ilk çalışmalar TKA mimarileri üzerine olmuştur. Ancak TKA'ların sadece doğrusal problemleri öğrenebildiğinin anlaşılması üzerine, doğrusal olmayan problemleri de öğrenebilen ÇKA mimarisi doğmuştur. ÇKA'lar girdi ve çıktı katmanının yanı sıra ara (gizli) katman

adı verilen bir katmana daha sahiptir. ÇKA mimarisinin öğrenme kuralı, ileriye doğru hesaplama ve geriye doğru hesaplama olarak ikiye ayrılır. İleriye doğru hesaplama ağın çıktısını oluşturma, geriye doğru hesaplama ise ağırlıkları güncelleme safhasıdır [2]. Şekil 2'de ÇKA mimarisinin görüntüsü verilmiştir.

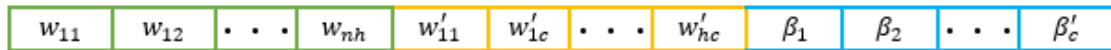


Şekil 2. Çok katmanlı algılayıcı modeli

YSA mimarisi eğitilirken, ileri yayılım yaptıktan sonra veri setindeki gerçek sınıf etiketleri ve tahmin edilen sınıf etiketlerine bakar. Buna göre her bir örneklem için bir hata değeri hesaplar. Bu çalışmada, hata hesabı için Eşitlik (1)'de verilen MSE yöntemi kullanılmıştır.

$$MSE = \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^n (y_j - \hat{y}_j)^2 \quad (1)$$

Burada  $m$  eğitim setindeki örnek sayısını,  $n$  sınıf etiketi sayısını,  $y_j$  çıktı için beklenen değeri ve  $\hat{y}$  ise  $j$ . çıktı için YSA tarafından tahmin edilen değeri ifade etmektedir.



Şekil 3. Ağırlıkların vektör olarak modellenmesi

Vektör uzunlukları veri setindeki öznitelik sayısına, ara katmandaki nöron sayısına ve veri setinin sınıf sayısına göre değişmektedir. Öznitelik ve sınıf sayısı veri setine bağlı olduğu için sabittir ve değiştirilemez. Ara katmandaki nöron sayısı ise

### 2.3. YSA ve Metasezgisel Algoritmaların Hibrit Kullanımı

İleri yayılım aşamasında hesaplanan MSE değerinin en aza indirilmesi için ağırlıkların güncellenmesi gerekir. Geri yayılım olarak adlandırılan bu aşama için standart YSA'da türev tabanlı teknikler kullanılır. Çalışmanın Giriş bölümünde detaylıca bahsedildiği üzere türev tabanlı teknikler yerel minimuma takılma sorunu yaşayabilmektedir. Bu sorunu ortadan kaldırmak için ağırlıkların güncellenmesi aşamasında metasezgisel algoritmalarından faydalanılmıştır. Şekil 3'te YSA'daki ağırlıkların vektör olarak modellenmesi görülmektedir.

YSA mimarisini oluşturan kullanıcının tercihi göre tasarlanabilir. Bu çalışmada standart olması açısından ara katmandaki nöron sayısı Eşitlik (2)'ye göre belirlenmiştir.

$$araKatmandakiNöronSayisi = (oznitelikSayisi \times 2) + 1 \quad (2)$$

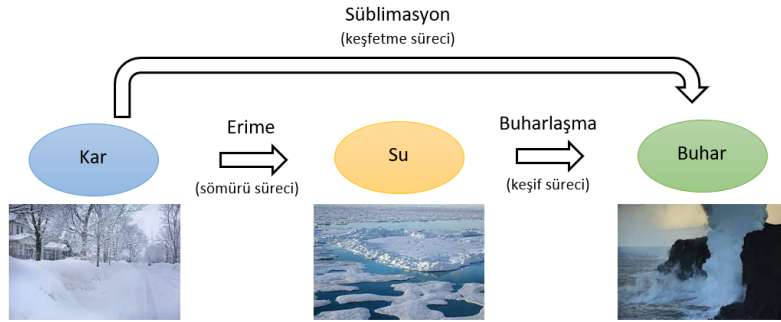
Bu vektörler metasezgisel algoritmalar tarafından üretilen aday çözümleri temsil eder. Vektördeki ağırlıkların güncellenmesi işlemi, seçilen metasezgisel algoritmanın konum güncelleme stratejisine bağlı olarak gerçekleştirilecektir. Metasezgisel algoritmadaki her bireyin uygunluğu, o bireye ait ağırlık vektörünün ileri yönü hareketi sonucunda hesaplanan *MSE* değerine eşittir. Sonuç olarak, ileri yayılım aşamasında standart YSA yapısından, hatanın geri yayılımı aşamasında ise seçilen metasezgisel algoritmanın işlem adımlarından faydalanan hibrit bir model geliştirilmiş olur.

Bu araştırma makalesinde, 2023 yılında önerilen SAO algoritması kullanılarak hibrit bir YSA modeli geliştirilmiştir. Bu model sayesinde SAO algoritmasının yerel minimumlardan kurtulmak için

sahip olduğu stratejilerden YSA modelinin de faydalanması sağlanmıştır.

#### 2.4. Kar Erime Optimizasyonu Algoritması

Kar Erime Optimizasyonu (snow ablation optimizer, SAO) algoritması, karın süblimleşme ve erime davranışını taklit eden metasezgisel bir algoritmadır. Şekil 4'te gösterildiği gibi kar, fiziksel olarak sıvı su ve buhar olmak üzere iki forma dönüşebilmektedir. Bu iki forma karşılık gelen süreçler de erime ve süblimleşme sürecidir veya eriyen su buharlaşarak buhar haline de gelebilmektedir [25]. Süblimleşme eylemi, katı maddelerin ısı ile eriyip sıvı hale geçmeden gaz hale geçmesi anlamına gelmektedir. SAO, karın erime ve süblimleşme süreçlerini taklit eder.



Şekil 4. Algoritmanın geliştirilmesinde ilham alınan kar erime olayı [25]

Metasezgisel algoritmalarındaki en büyük sorunlardan biri keşif ve sömürü arası oluşan dengesizlik halidir. SAO bu iki durumu dengelemek için oluşturulmuş bir algoritmadır [25]. SAO algoritmasının işleyişinde yer alan i) *başlatma*, ii) *keşif* ve iii) *sömürü* aşamaları ve *ikili popülasyon oluşturma* mekanizması aşağıda anlatılmaktadır.

i) *Başlatma aşaması*: popülasyon Eşitlik (3) kullanılarak arama uzayı sınırları içinde rastgele başlatılır. Burada  $Z$ , popülasyonu temsil eden bir matristir.  $N$  popülasyondaki ajan sayısını,  $Dim$  problemin boyutunu,  $L$  arama uzayının alt sınır değerini,  $U$  arama uzayının üst sınır değerini ve  $\theta$  ise  $[0, 1]$  aralığında rastgele sayılardan oluşan  $N \times Dim$  boyutlu bir matrisi ifade eder.

$$Z = L + \theta \times (U - L) = \begin{bmatrix} Z_{1,1} & Z_{1,2} & \dots & Z_{1,Dim-1} & Z_{1,Dim} \\ Z_{2,1} & Z_{2,2} & \dots & Z_{2,Dim-1} & Z_{2,Dim} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ Z_{N-1,1} & Z_{N-1,2} & \dots & Z_{N-1,Dim-1} & Z_{N-1,Dim} \\ Z_{N,1} & Z_{N,2} & \dots & Z_{N,Dim-1} & Z_{N,Dim} \end{bmatrix}_{N \times Dim} \quad (3)$$

ii) Keşif aşaması: kar veya su buharlaştığında düzensiz bir şekilde dağılmaya başlar. Eşitlik (4)'te verilen Brownian hareketi bu düzensiz dağılımı taklit eder.

$$f_{BM}(x; 0,1) = \frac{1}{\sqrt{2\pi}} \times \exp\left(-\frac{x^2}{2}\right) \quad (4)$$

$$Z_i(t+1) = Elite(t) + BM_i(t) \otimes (\theta_1 \times (G(t) - Z_i(t)) + (1 - \theta_1) \times (Z(t) - Z_i(t))) \quad (5)$$

Burada,  $i = 1, 2, \dots, N$  konumu güncellenen ajanın indisini,  $t$  o anki iterasyonu,  $BM_i(t)$  Brownian hareketini,  $\otimes$  girdilerin çarpımını,  $\theta_1$   $[0,1]$  aralığında rastgele oluşturulan bir sayıyı,  $G(t)$  o ana kadar elde edilen en iyi çözümü,  $Z(t)$  popülasyonun sütun bazında ağırlıklarının ortalamasını içeren vektörü temsil etmektedir. En iyi uygunluğa sahip üç birey ve en iyi uygunluğa sahip  $N/2$  bireyin

Popülasyondaki arama ajanlarının konumları Brownian hareketinden faydalanılarak Eşitlik (5)'teki gibi güncellenir.

konumlarının ortalaması *EliteHavuzu* adı verilen bir matriste tutulur. *Elite(t)*, *EliteHavuzu* içinden rastgele seçilen bir çözümü ifade eder.

iii) Sömürü aşaması: karın su haline geçme davranışını taklit eder. Ajanlarının konumları, popülasyondaki en iyi çözümlerin etkisiyle Eşitlik (6) kullanılarak güncellenir.

$$Z_i(t+1) = M \times G(t) + BM_i(t) \otimes (\theta_2 \times (G(t) - Z_i(t)) + (1 - \theta_2) \times (Z(t) - Z_i(t))) \quad (6)$$

Burada,  $M$  parametresi kar erime oranını,  $\theta_2$   $[-1, 1]$  aralığında rastgele bir sayıyı ifade etmektedir.

SAO algoritmasının tam konum güncelleme denklemi Eşitlik (7)'de verilmiştir.

İkili popülasyon oluşturma aşaması: bu aşamada keşif ve sömürü arasında denge oluşturmak amaçlanır. Su haline gelen karın bir kısmı arama esnasında buharlaşarak merkezîyetçiliğini kaybeder. Bu dengenin sağlanması için ikili popülasyon yapısı oluşturulmuştur. Popülasyon arama sürecinin başlarında eşit sayıda arama ajanından oluşacak şekilde rastgele iki alt popülasyona ayrılır. Alt popülasyonlardan ilki keşiften, ikincisi sömürüden sorumludur. Her iterasyonda ikinci alt popülasyonun ajan sayısı birer azaltılırken, ilkinin ajan sayısı birer artırılır. Bu mekanizmanın sözde kodu Şekil 5'te sunulmuştur.

İkili popülasyon oluşturma mekanizması
$N$ : popülasyondaki ajan sayısı
Başlatma: $t = 0, t_{\max}, N_a = N_b = N/2$
<b>While</b> ( $t < t_{\max}$ )
<b>If</b> $N_a < N$
$N_a = N_a + 1$
$N_b = N_b - 1$
<b>End if</b>
$t = t + 1$
<b>End while</b>

Şekil 5. İkili popülasyon oluşturma mekanizması

$$Z_i(t+1) = \begin{cases} Elite(t) + BM_i(t) \otimes (\theta_1 \times (G(t) - Z_i(t)) \\ \quad + (1 - \theta_1) \times (Z(t) - Z_i(t))), i \in index_a \\ M \times G(t) + BM_i(t) \otimes (\theta_2 \times (G(t) - Z_i(t)) \\ \quad + (1 - \theta_2) \times (Z(t) - Z_i(t))), i \in index_b \end{cases} \quad (7)$$

Eşitlik (3)'te açıklandığı gibi, tüm popülasyon bir konum matrisidir. Dolayısıyla Eşitlik (7)'deki  $index_a$  ve  $index_b$ , tüm konum matrisinde alt

popülasyonlardaki bireylerin satır numaralarını içeren bir dizi indeksi belirtir. SAO algoritmasının sözde kodu Şekil 6'da sunulmuştur.

```
Başlangıç için sürüdeki birey sayısı, ağırlık, tolerans değeri belirlenir:  
 $Z_i (i = 1, 2, \dots, N), t = 0, t_{max}, N_a = N_b = \frac{N}{2}$   
Uygunlukları hesaplanır.  
Mevcut en iyi birey kaydedilir.  
While ( $t < t_{max}$ )  
    Kar erime hızı hesaplanır.  
    Tüm popülasyon rastgele iki alt popülasyona ayrılır.  
    ForEach birey  
        Eşitlik 7 ile her ajanın konumu güncellenir.  
    End foreach  
    Uygunluk fonksiyonu değerlendirme yapılır.  
    En iyi ajan güncellenir.  
End while  
En iyi ajan alınır.
```

Şekil 6. SAO algoritmasının sözde kodu [25]

## 2.5. Karşılaştırma Algoritmaları

### 2.5.1. Guguk Kuşu Arama (CS) Algoritması

CS algoritması, Yang ve arkadaşları [44] tarafından geliştirilmiş bir algoritmadır. Bu algoritma, guguk kuşlarının yuva parazitliğine dayanır [45]. Yang ve arkadaşları [44] bu algoritmayı, aşağıdaki üç kuralla açıklamışlardır:

- Dişi guguk kuşu öncelikle rastgele bir yuva seçerek yumurtasını bırakır.
- En kaliteli yumurtalara sahip, en iyi yuvalar bir sonraki nesillere aktarılır.
- Ev sahibi kuş genellikle guguk kuşu tarafından bırakılan yumurtayı tespit ederek konakçı yumurtayı yuvasından atar ya da kendi yuvasını terk ederek yeni bir yuva kurar.

### 2.5.2. Gri Kurt Optimizasyonu (GWO) Algoritması

GWO algoritması, Mirjalili ve arkadaşları [46] tarafından, gri kurtların avlanma ve liderlik özelliklerinden ilham alınarak oluşturulmuş metasezgisel bir algoritmadır. Alfa, beta, delta, omega olarak isimlendirilen dört tür gri kurt ile liderlik özellikleri simüle edilmiştir [46]. Alfa en iyi, beta ikinci en iyi, delta üçüncü en iyi çözümleri ifade ederken, omega popülasyonun geriye kalan çözümlerini ifade eder.

### 2.5.3. Sürüngen Arama Algoritması (RSA)

RSA, Abualigah ve arkadaşları [47] tarafından timsahların davranışlarından ilham alınarak geliştirilmiş metasezgisel bir algoritmadır. Bu algoritmanın iki ana aşaması vardır. Bunlar, avın kuşatılması ve avlanması olmaktadır. Bu iki aşamanın da ikişer arama stratejisi vardır. Keşif aşamasına ait iki strateji, yüksek yürüyüş ve göbek yürüyüşü arama yöntemleridir. Sömürü aşamasının stratejileri ise av koordinasyonu ve av iş birliğidir.

### 2.5.4. Sinüs Kosinüs Algoritması (SCA)

SCA, Mirjalili [48] tarafından, matematikte aynı isme sahip olan sinüs ve kosinüs fonksiyonlarından ilham alınarak önerilmiş metasezgisel bir algoritmadır. Keşif aşaması için sinüs, sömürü aşaması için kosinüs fonksiyonu kullanılmıştır.

## 2.6. Karmaşıklık Matrisi (Confusion Matrix)

Karmaşıklık matrisi, modelin doğruluğunu, veri setlerinin sınıflandırma başarısını ölçmeye yarayan ve literatürde sıklıkla kullanılan bir performans ölçüm yöntemidir. Karmaşıklık matrisinde tahmin edilen sınıf etiketi ve gerçek sınıf etiketi üzerinden Şekil 7'de verilen durumlara göre çeşitli hesaplamalar yapılır.



		Tahmin	
		Negatif	Pozitif
Gerçek Durum	Negatif	Doğru Negatif (True Negative, TN)	Yanlış Pozitif (False Pozitive, FP)
	Pozitif	Yanlış Negatif (False Negative, FN)	Doğru Pozitif (True Pozitive, TP)

Şekil 7. Karmaşıklık matrisi

Tahmini sınıf *pozitif* ve gerçek sınıf *pozitif* ise doğru pozitif (True Positive, TP), tahmini sınıf *negatif* ve gerçek sınıf *negatif* ise doğru negatif (True Negative, TN), tahmini sınıf *pozitif* ve gerçek sınıf *negatif* ise yanlış pozitif (False Positive FP), tahmini sınıf *negatif* ve gerçek sınıf *pozitif* ise yanlış negatif (False Negative, FN) olarak isimlendirilir. Bu bilgiler kullanılarak aşağıdaki metrikler elde edilmiştir.

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{F1-score} = 2 * \frac{\text{Sensitivity} * \text{Precision}}{\text{Sensitivity} + \text{Precision}}$$

- Sensitivity, doğru pozitif değerlerin toplam gerçek pozitif değerlerine bölünmesi ile elde edilir. Pozitif sınıfa ait değerlerden doğru tahminlerin oranının ne olduğu hakkında bilgi verir.

- Specificity, gerçek negatif değerlerin, toplam gerçek negatif değerlere bölünmesi ile elde edilir.
- Precision, doğru pozitif değerlerin toplam tahmini pozitif değerlerine bölünmesiyle elde edilir. Bu metrik, doğru tahminlerin ne oranda olduğu hakkında bilgi verir.
- F1-score ise, sensitivity ve precision metriklerinin harmonik ortalamasıdır. Eşit dağılmayan veri setleri için hatalı modeller oluşturmanın önüne geçmesi bakımından önemli bir metriktir.

Tüm metrikler en kötü 0, en iyi 1 değerini alırlar.

### 3. DENEYSEL ÇALIŞMALAR

Bu çalışmada geliştirilen SAO hibrit modeli, CS, GWO, RSA ve SCA algoritmaları ile geliştirilen hibrit modellerle karşılaştırılmıştır. Deneysel çalışmalar aggregation, balance, liver, pathbased ve wine veri setleri üzerinde yapılmıştır. Veri setleri %75 eğitim, %25 test verisi olarak ayrılmıştır. Ağırlık değerleri [-10, 10] arama uzayında oluşturulmuştur. Her bir yöntem, her bir veri setinde 50 birey ve 400 iterasyon için 30 kez çalıştırılmıştır. Elde edilen sonuçlar için hesaplanan sensitivity, specificity, precision ve f1-score metriklerinin ortalama, standart sapma ve başarı sırası değerleri verilmiştir. Ayrıca çizelgelerin en alt satırında, yöntemlerin ortalama başarı sırası (*OBS*) verilmiştir.

Çizelge 2. SAO, CS, GWO, RSA ve SCA algoritmalarının sensitivity metriğine göre ortalama, standart sapma ve başarı sırası değerleri

Veri seti	SAO			GWO			RSA			CS			SCA		
	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra
aggregation	0,355	0,122	1	0,340	0,121	2	0,220	0,058	5	0,294	0,050	3	0,257	0,031	4
balance	0,684	0,084	1	0,605	0,017	2	0,542	0,083	5	0,604	0,037	3	0,565	0,073	4
liver	0,396	0,033	1	0,377	0,034	3	0,355	0,037	5	0,377	0,031	2	0,359	0,032	4
pathbased	0,600	0,076	1	0,540	0,053	3	0,384	0,160	5	0,554	0,065	2	0,458	0,155	4
wine	0,683	0,077	2	0,701	0,119	1	0,594	0,145	5	0,663	0,057	3	0,638	0,059	4
<i>OBS</i>			1,2 (1)			2,2 (2)			5 (5)			2,6 (3)			4 (4)

Çizelge 2’deki sensitivity sonuçları incelendiğinde, SAO hibrit modelinin 5 veri setinin 4’ünde ilk sırayı elde ettiği, wine veri setinde ise GWO hibrit

modelinin ardından 2.sırada yer aldığı gözlenmektedir. OBS değerlerine bakıldığında SAO hibrit modelinin 1.2 ortalama ile diğer

algoritmaların üstün bir performansa sahip olduğu değerlerine bakıldığında ise başarı sıraları GWO görülmektedir. Karşılaştırma modellerinin OBS (2.2), CS (2.6), SCA (4) ve RSA (5) şeklindedir.

**Çizelge 3.** SAO, CS, GWO, RSA ve SCA algoritmalarının specificity metriğine göre ortalama, standart sapma ve başarı sırası değerleri

Veri seti	SAO			GWO			RSA			CS			SCA		
	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra
aggregation	0,911	0,025	1	0,910	0,026	2	0,883	0,021	5	0,903	0,008	3	0,895	0,011	4
balance	0,919	0,016	1	0,899	0,014	2	0,845	0,071	5	0,895	0,026	3	0,865	0,062	4
liver	0,729	0,032	1	0,710	0,034	3	0,688	0,036	5	0,711	0,030	2	0,693	0,031	4
pathbased	0,795	0,039	1	0,764	0,025	3	0,685	0,082	5	0,772	0,031	2	0,723	0,0789	4
wine	0,855	0,031	2	0,863	0,052	1	0,800	0,076	5	0,845	0,025	3	0,833	0,0310	4
OBS			1,2 (1)			2,2 (2)			5 (5)			2,6 (3)			4 (4)

Çizelge 3'teki specificity sonuçları incelendiğinde, sensitivity metriğiyle aynı performans sonuçlarına ulaşıldığı gözlenmiştir. SAO hibrit modelinin 5 veri setinin 4'ünde ilk sırayı elde ettiği, wine veri setinde GWO hibrit modelinin ardından 2.sırada yer aldığı ve OBS açısından 1.2 ortalama ile diğer

algoritmaların daha iyi bir performansa sahip olduğu görülmektedir. Karşılaştırma modellerinin OBS değerlerine bakıldığında ise başarı sıraları GWO (2.2), CS (2.6), SCA (4) ve RSA (5) şeklindedir.

**Çizelge 4.** SAO, CS, GWO, RSA ve SCA algoritmalarının precision metriğine göre ortalama, standart sapma ve başarı sırası değerleri

Veri seti	SAO			GWO			RSA			CS			SCA		
	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra
aggregation	0,301	0,111	1	0,296	0,110	2	0,155	0,059	5	0,233	0,048	3	0,197	0,029	4
balance	0,670	0,100	1	0,578	0,009	3	0,543	0,033	5	0,599	0,090	2	0,559	0,031	4
liver	0,431	0,038	1	0,378	0,098	3	0,287	0,117	5	0,407	0,066	2	0,347	0,104	4
pathbased	0,647	0,116	1	0,527	0,131	3	0,334	0,202	5	0,624	0,095	2	0,473	0,228	4
wine	0,591	0,135	2	0,606	0,171	1	0,513	0,212	5	0,566	0,114	3	0,528	0,084	4
OBS			1,2 (1)			2,4 (2)			5 (4)			2,4 (2)			4 (3)

Çizelge 4'teki precision sonuçları incelendiğinde, SAO hibrit modelinin 5 veri setinin 4'ünde ilk sırayı elde ettiği, wine veri setinde GWO hibrit modelinin ardından 2.sırada yer aldığı gözlenmektedir. OBS değerlerine bakıldığında SAO hibrit modelinin 1.2 ortalama ile diğer algoritmaların üstün bir

performansa sahip olduğu görülmektedir. Karşılaştırma modellerinin OBS değerlerine bakıldığında GWO (2.4) ve CS (2.4) aynı başarı sırasıyla SAO modelinin ardından 2. olmuşlardır. SCA (3) ve RSA (4) modelleri ise performans açısından diğer modellerin gerisinde kalmıştır.

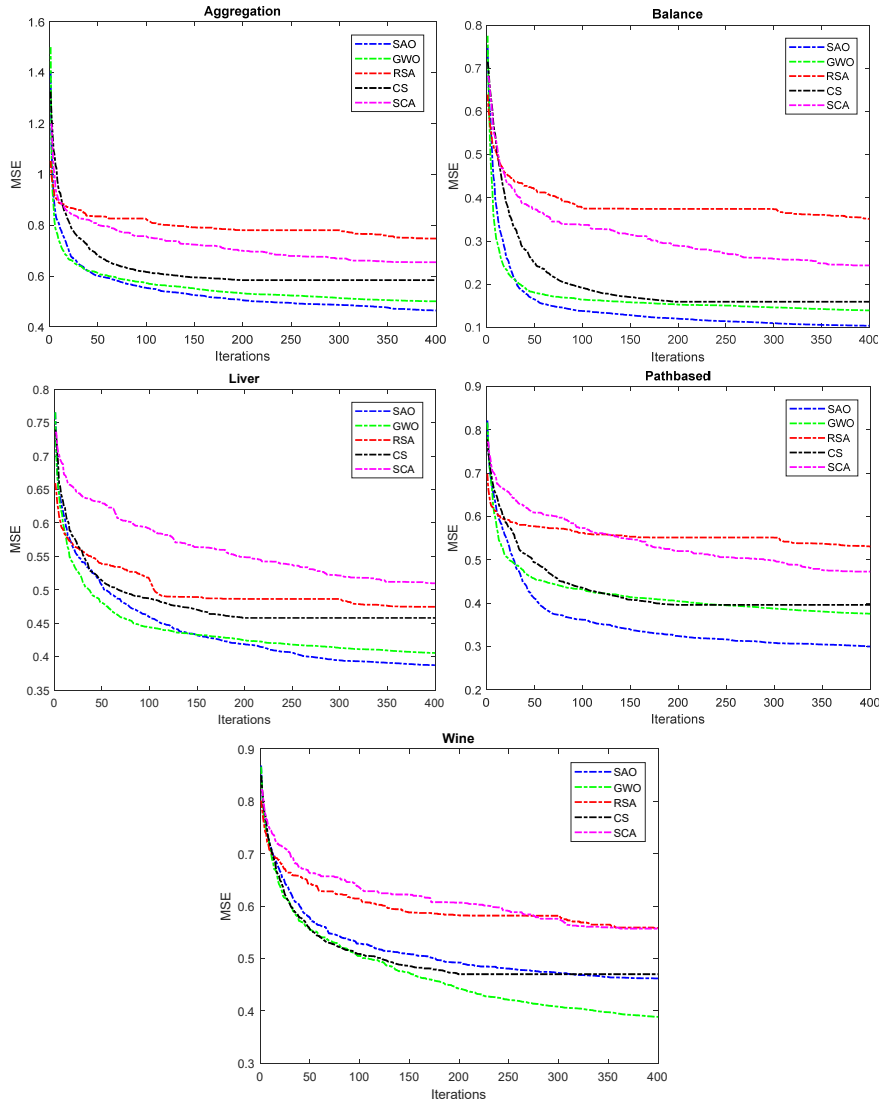
**Çizelge 5.** SAO, CS, GWO, RSA ve SCA algoritmalarının fl-score metriğine göre ortalama, standart sapma ve başarı sırası değerleri

Veri seti	SAO			GWO			RSA			CS			SCA		
	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra
aggregation	0,294	0,120	1	0,288	0,116	2	0,153	0,057	5	0,237	0,041	3	0,200	0,029	4
balance	0,663	0,083	1	0,582	0,016	3	0,509	0,100	5	0,582	0,041	2	0,536	0,080	4
liver	0,377	0,049	1	0,342	0,064	3	0,290	0,073	5	0,350	0,055	2	0,318	0,062	4
pathbased	0,525	0,103	1	0,456	0,046	3	0,305	0,135	5	0,467	0,077	2	0,372	0,132	4
wine	0,613	0,097	2	0,634	0,148	1	0,514	0,175	5	0,587	0,071	3	0,560	0,069	4
OBS			1,2 (1)			2,4 (2)			5 (4)			2,4 (2)			4 (3)

Çizelge 5'teki f1-score sonuçları incelendiğinde, precision metriğiyle aynı performans sonuçlarına ulaşıldığı gözlenmiştir. SAO hibrit modelinin 5 veri setinin 4'ünde ilk sırayı elde ettiği, wine veri setinde GWO hibrit modelinin ardından 2.sırada yer aldığı, OBS değerlerine bakıldığında ise SAO hibrit modelinin 1.2 ortalama ile diğer algoritalardan üstün bir performansa sahip olduğu görülmektedir. Karşılaştırma modellerinin OBS değerlerine bakıldığında GWO (2.4) ve CS (2.4) aynı başarı sırasıyla SAO modelinin ardından 2. olmuşlardır.

SCA (3) ve RSA (4) modelleri ise performans açısından diğer modellerin gerisinde kalmıştır.

Şekil 8'de ise hibrit modellerin eğitim aşamasındaki MSE değerleri üzerinden hata yakınsama grafikleri çizdirilmiştir. Yakınsama grafikleri incelendiğinde SAO hibrit modelinin wine veri setinde GWO hibrit modelinin ardından ikinci olduğu, diğer tüm veri setlerinde ise en hızlı yakınsama performansını elde ettiği gözlenmiştir.



Şekil 8. MSE yakınsama grafikleri

### 3.1. Önerilen Hibrit Modelde Popülasyon Genişliğinin Etkisi

Bu bölümde, çalışmada önerilen SAO hibrit modelinin başarısında metasezgisel algoritmanın sahip olduğu parçacık sayısının etkisi incelenmiştir. Bu amaçla SAO'daki parçacık sayısı 10, 20, 30, 40, 50, 100, 150 ve 200 olarak ayarlanmış ve adil karşılaştırma yapmak amacıyla her bir model 20000 uygunluk sayısı için çalıştırılmıştır. Modeller tarafından elde edilen metrik değerleri, başarı sıraları ve OBS değerleri Çizelge 6-9'da sunulmuştur.

Çizelge 6, 7 ve 8'deki sensitivity, specificity ve precision metrik değerleri incelendiğinde, parçacık sayısı 100 olduğunda 2.8 OBS değeri ile en başarılı sonuçlara ulaşıldığı gözlenmiştir. Çizelge 9'da verilen F1-score metriğinde ise parçacık sayısı 40 olduğunda 2.8 OBS değeri ile en başarılı sonuçlar elde edilirken, parçacık sayısının 30, 50 ve 100 olarak ayarlandığı durumlarda 3.2 OBS değeri elde edilmiştir. Elde edilen sonuçlar birlikte değerlendirildiğinde, parçacık sayısının 10, 20 gibi düşük değerler ya da 150, 200 gibi yüksek değerler almasının modelin başarısını olumsuz etkilediği, 50, 100 gibi değerlerin ise modelin başarısını artırdığı görülmektedir.

**Çizelge 6.** SAO hibrit modelinin 10, 20, 30, 40, 50, 100, 150 ve 200 parçacık sayıları için elde ettiği sensitivity metrik değerleri

Parçacık sayısı	10			20			30			40		
	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra
aggregation	0,329	0,103	7	0,333	0,089	6	0,362	0,100	2	0,333	0,095	5
balance	0,644	0,050	7	0,684	0,076	1	0,679	0,090	4	0,681	0,087	3
liver	0,390	0,039	5	0,389	0,033	6	0,381	0,036	8	0,401	0,031	1
pathbased	0,529	0,093	7	0,578	0,087	5	0,616	0,100	1	0,579	0,091	4
wine	0,659	0,070	6	0,679	0,089	5	0,712	0,116	1	0,693	0,094	3
<i>OBS</i>			6,4			4,6			3,2			3,2
Parçacık sayısı	50			100			150			200		
	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra
aggregation	0,355	0,122	4	0,391	0,107	1	0,356	0,085	3	0,310	0,081	8
balance	0,684	0,084	2	0,670	0,076	5	0,666	0,090	6	0,604	0,040	8
liver	0,396	0,033	4	0,398	0,026	3	0,399	0,027	2	0,386	0,033	7
pathbased	0,600	0,076	2	0,585	0,062	3	0,399	0,027	8	0,567	0,056	6
wine	0,683	0,077	4	0,693	0,097	2	0,399	0,027	8	0,650	0,063	7
<i>OBS</i>			3,2			2,8			5,4			7,2

**Çizelge 7.** SAO hibrit modelinin 10, 20, 30, 40, 50, 100, 150 ve 200 parçacık sayıları için elde ettiği specificity metrik değerleri

Parçacık sayısı	10			20			30			40		
	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra
aggregation	0,905	0,021	8	0,906	0,021	5	0,911	0,023	3	0,906	0,022	6
balance	0,913	0,015	6	0,921	0,020	1	0,919	0,022	4	0,920	0,018	2
liver	0,724	0,039	5	0,722	0,033	6	0,715	0,036	8	0,734	0,031	1
pathbased	0,761	0,047	7	0,784	0,045	5	0,805	0,052	1	0,785	0,046	4
wine	0,843	0,032	6	0,851	0,038	5	0,867	0,048	1	0,859	0,039	3
<i>OBS</i>			6,4			4,4			3,4			3,2
Parçacık sayısı	50			100			150			200		
	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra
aggregation	0,911	0,026	2	0,921	0,022	1	0,911	0,018	4	0,905	0,021	7
balance	0,919	0,017	3	0,916	0,017	5	0,912	0,023	7	0,895	0,020	8
liver	0,729	0,033	4	0,731	0,026	3	0,732	0,027	2	0,720	0,033	7
pathbased	0,796	0,040	2	0,787	0,032	3	0,732	0,027	8	0,778	0,028	6
wine	0,855	0,031	4	0,860	0,044	2	0,732	0,027	8	0,838	0,032	7
<i>OBS</i>			3			2,8			5,8			7

**Çizelge 8.** SAO hibrit modelinin 10, 20, 30, 40, 50, 100, 150 ve 200 parçacık sayıları için elde ettiği precision metrik değerleri

Parçacık sayısı	10			20			30			40		
	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra
aggregation	0,272	0,103	7	0,276	0,078	6	0,313	0,115	2	0,291	0,084	5
balance	0,654	0,108	6	0,683	0,098	1	0,663	0,097	5	0,667	0,099	4
liver	0,420	0,093	6	0,430	0,063	4	0,405	0,077	8	0,426	0,066	5
pathbased	0,536	0,160	7	0,607	0,130	5	0,692	0,101	1	0,617	0,137	3
wine	0,565	0,128	7	0,578	0,139	5	0,632	0,169	1	0,623	0,159	2
<i>OBS</i>			6,6			4,2			3,4			3,8
Parçacık sayısı	50			100			150			200		
	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra
aggregation	0,302	0,111	3	0,338	0,105	1	0,295	0,089	4	0,254	0,056	8
balance	0,670	0,100	3	0,673	0,109	2	0,645	0,104	7	0,580	0,031	8
liver	0,432	0,038	3	0,445	0,031	2	0,448	0,036	1	0,408	0,062	7
pathbased	0,647	0,116	2	0,598	0,116	6	0,448	0,036	8	0,616	0,116	4
wine	0,591	0,136	4	0,596	0,148	3	0,448	0,036	8	0,570	0,119	6
<i>OBS</i>			3			2,8			5,6			6,6

**Çizelge 9.** SAO hibrit modelinin 10, 20, 30, 40, 50, 100, 150 ve 200 parçacık sayıları için elde ettiği fl-score metrik değerleri

Parçacık sayısı	10			20			30			40		
	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra
aggregation	0,268	0,105	7	0,274	0,083	5	0,298	0,104	2	0,270	0,089	6
balance	0,632	0,065	7	0,672	0,081	1	0,662	0,091	4	0,665	0,088	2
liver	0,360	0,069	7	0,366	0,055	6	0,353	0,062	8	0,383	0,051	1
pathbased	0,444	0,111	7	0,495	0,110	5	0,548	0,138	1	0,502	0,115	3
wine	0,583	0,091	6	0,607	0,112	5	0,651	0,141	1	0,633	0,117	2
<i>OBS</i>			6,8			4,4			3,2			2,8
Parçacık sayısı	50			100			150			200		
	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra	ort.	std.	sıra
aggregation	0,294	0,120	3	0,336	0,110	1	0,284	0,078	4	0,252	0,074	8
balance	0,663	0,083	3	0,655	0,078	5	0,640	0,086	6	0,581	0,034	8
liver	0,378	0,049	4	0,381	0,036	3	0,382	0,044	2	0,367	0,052	5
pathbased	0,525	0,104	2	0,501	0,079	4	0,382	0,044	8	0,478	0,069	6
wine	0,614	0,097	4	0,623	0,122	3	0,382	0,044	8	0,579	0,084	7
<i>OBS</i>			3,2			3,2			5,6			6,8

#### 4. TARTIŞMA VE SONUÇ

Bu çalışmada, yakın zamanda önerilen SAO algoritması ile YSA'daki ağırlık değerlerinin eğitilmesi için hibrit bir model geliştirilmiştir. Geliştirilen model 5 farklı veri seti (aggregation, balance, liver, pathbased ve wine) üzerinde 4 farklı metasezgisel algoritma (GWO, RSA, CS ve SCA) ile oluşturulan hibrit modeller ile karşılaştırılmıştır. Modellerin başarısı sensitivity, specificity, precision ve fl-score metrikleri üzerinden karşılaştırılmıştır. SAO algoritmasının dört metriğin tamamında ortalama başarı sırası açısından en iyi performansı elde ettiği gözlenmiştir. Ayrıca

hibrit modellerin, eğitim aşamasındaki ortalama karesel hata değerlerinin yakınsama grafikleri çizdirilmiş ve SAO hibrit modelinin wine hariç diğer tüm veri setlerinde karşılaştırılan modellerden daha hızlı bir yakınsama performansına sahip olduğu gözlenmiştir. Parçacık sayısı ile ilgili yapılan deneysel çalışmalar ise 100 birey kullanılması durumunda SAO modelinin başarısının arttığını göstermiştir. Özetle, bu çalışmada elde edilen bulgular, SAO algoritmasının YSA'daki ağırlık değerlerinin güncellenmesinde önemli bir başarıya sahip olduğu gözlenmiştir. SAO'nun elde ettiği başarıda yerel minimumlardan kurtulma stratejilerinin önemli bir payı olduğu

düşünülmektedir. Bu sebeple ileriki çalışmalarda SAO'nun; derin öğrenme, sınıflandırma, kümeleme gibi yerel minimumlara sahip çeşitli gerçek dünya problemlerinin çözümünde tek başına veya hibrit olarak kullanılması önerilmektedir. Ayrıca sürekli problemlerin yanı sıra SAO'nun ikili veya kombinatoryal problemlerin çözümüne uyarlanması ve performansının test edilmesi de önemli katkılar sağlayabilir. Son olarak SAO'nun wine veri setinde diğer veri setlerinde olduğu kadar başarılı olamaması dikkat çekicidir. Bu duruma wine veri seti için oluşturulan modelin boyut sayısının diğer veri setleri için oluşturulan modellerden daha çok boyuta sahip olması yol açmış olabilir. SAO'nun yüksek boyutlu problemlerde performansının iyileştirilmesi de ayrı bir araştırma konusu olabilir.

## 5. KAYNAKÇA

1. Turing, A.M., 2009. Computing Machinery and Intelligence. Springer.
2. Öztemel, E., 2003. Yapay Sinir Ağları. Papatya Yayıncılık.
3. Rosenblatt, F., 1958. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological review, 65(6), 386.
4. Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning Representations by Back-Propagating Errors. Nature, 323(6088), 533-536.
5. Ciregan, D., Meier, U., Schmidhuber, J., 2012. Multi-Column Deep Neural Networks for Image Classification. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 3642-3649.
6. Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, 25, 1-9.
7. Liu, X., Zeng, S., Namaiti, A., Xin, R., 2023. Comparison between Three Convolutional Neural Networks for Local Climate Zone Classification Using Google Earth Images: A Case Study of the Fujian Delta in China. Ecological Indicators, 148, 110086.
8. Bas, E., Egrioglu, E., Cansu, T., 2024. Robust Training of Median Dendritic Artificial Neural Networks for Time Series Forecasting. Expert Systems with Applications, 238, 122080.
9. Dalal, A.-A., AlRassas, A.M., Al-qaness, M.A., Cai, Z., Aseeri, A.O., Abd Elaziz, M., Ewees, A.A., 2023. Tlia: Time-Series Forecasting Model Using Long Short-Term Memory Integrated with Artificial Neural Networks for Volatile Energy Markets. Applied Energy, 343, 121230.
10. Egrioglu, E., Baş, E., Chen, M.-Y., 2022. Recurrent Dendritic Neuron Model Artificial Neural Network for Time Series Forecasting. Information Sciences, 607, 572-584.
11. Ergun, U., Tayfun, D., 2020. Jaya Algoritması ile Optimize Edilmiş Yapay Sinir Ağlarını Kullanarak Türkiye'de Elektrik Enerjisi Tüketiminin Tahmini. Gazi University Journal of Science Part C: Design Technology, 8(3), 511-528.
12. Jayasimha, S., Lingaraju, K., Raju, H., 2022. Prediction of Surface Finish in Extrusion Honing Process by Regression Analysis and Artificial Neural Networks. Applications in Engineering Science, 10, 100105.
13. Selim, A., Shuvo, S.N.A., Moniruzzaman, M., Islam, M., Shah, S., Ohiduzzaman, M., 2023. Predictive Models for Dissolved Oxygen in an Urban Lake by Regression Analysis and Artificial Neural Network. 1-11.
14. Tam, V.W., Butera, A., Le, K.N., Da Silva, L.C., Evangelista, A.C., 2022. A Prediction Model for Compressive Strength of Co2 Concrete Using Regression Analysis and Artificial Neural Networks. Construction Building Materials, 324, 126689.
15. Ghiassi, M., Burnley, C., 2010. Measuring Effectiveness of a Dynamic Artificial Neural Network Algorithm for Classification Problems. Expert Systems with Applications, 37(4), 3118-3128.
16. Xu, B., Su, J., Dale, D., Watson, M., 2000. Cotton Color Grading with a Neural Network. Textile Research Journal, 70(5), 430-436.
17. Yaman, S., Karakaya, B., Köküm, M., 2024. A Neural Network Approach for Classification of Fault-Slip Data in Geoscience. Ain Shams Engineering Journal, 15(1), 102325.

18. Elangasinghe, M., Singhal, N., Dirks, K., Salmond, J., Samarasinghe, S., 2014. Complex Time Series Analysis of Pm10 and Pm2.5 for a Coastal Site Using Artificial Neural Network Modelling and K-Means Clustering. *Atmospheric Environment*, 94, 106-116.
19. Erilli, N.A., Yolcu, U., Eğrioğlu, E., Aladağ, Ç.H., Öner, Y., 2011. Determining the Most Proper Number of Cluster in Fuzzy Clustering by Using Artificial Neural Networks. *Expert Systems with Applications*, 38(3), 2248-2252.
20. Nithya, A., Appathurai, A., Venkatadri, N., Ramji, D., Palagan, C.A., 2020. Kidney Disease Detection and Segmentation Using Artificial Neural Network and Multi-Kernel K-Means Clustering for Ultrasound Images. *Measurement*, 149, 106952.
21. Jiadong, Q., Ohl, J.P., Tran, T.-T., 2024. Predicting Clay Compressibility for Foundation Design with High Reliability and Safety: A Geotechnical Engineering Perspective Using Artificial Neural Network and Five Metaheuristic Algorithms. *Reliability Engineering System Safety*, 243, 109827.
22. Alameer, Z., Abd Elaziz, M., Ewees, A.A., Ye, H., Jianhua, Z., 2019. Forecasting Gold Price Fluctuations Using Improved Multilayer Perceptron Neural Network and Whale Optimization Algorithm. *Resources Policy*, 61, 250-260.
23. Du, W., Zhang, Q., Chen, Y., Ye, Z., 2021. An Urban Short-Term Traffic Flow Prediction Model Based on Wavelet Neural Network with Improved Whale Optimization Algorithm. *Sustainable Cities Society*, 69, 102858.
24. Ouladbrahim, A., Belaidi, I., Khatir, S., Magagnini, E., Capozucca, R., Wahab, M.A., 2022. Experimental Crack Identification of Api X70 Steel Pipeline Using Improved Artificial Neural Networks Based on Whale Optimization Algorithm. *Mechanics of Materials*, 166, 104200.
25. Deng, L., Liu, S., 2023. Snow Ablation Optimizer: A Novel Metaheuristic Technique for Numerical Optimization and Engineering Design. *Expert Systems with Applications*, 225, 120069.
26. Karakoyun, M., Özkış, A., 2021. Transfer Fonksiyonları Kullanarak İkili Güve-Alev Optimizasyonu Algoritmalarının Geliştirilmesi ve Performanslarının Karşılaştırılması. *Necmettin Erbakan Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, 3(2), 1-10.
27. Özkış, A., Karakoyun, M., 2023. A Binary Enhanced Moth Flame Optimization Algorithm for Uncapacitated Facility Location Problems. *Pamukkale University Journal of Engineering Sciences*, 29(7), 737-751.
28. Çelik, İ., Yıldız, C., Şekkel, M., 2018. Rüzgâr Enerji Santrali Kurulumunda Rüzgâr Türbinlerinin Mikro Yerleşimi İçin Bir Optimizasyon Modeli. *Gazi University Journal of Science Part C: Design Technology*, 6(4), 898-908.
29. Kong, M., Tian, P., Kao, Y., 2008. A New Ant Colony Optimization Algorithm for the Multidimensional Knapsack Problem. *Computers Operations Research*, 35(8), 2672-2683.
30. İrmak, B., 2022. Yapay Sinir Ağlarının Eğitimi İçin Kelebek Optimizasyonu Algoritmasının İyileştirilmesi. Yüksek Lisans Tezi, Necmettin Erbakan Üniversitesi, Fen Bilimleri Enstitüsü, Konya, Türkiye, 75.
31. Wolpert, D.H., Macready, W.G., 1997. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67-82.
32. Singh, P., Pandey, V.K., Chakraborty, S., Dash, K.K., Singh, R., Béla, K., 2023. Ultrasound-Assisted Extraction of Phytochemicals from Green Coconut Shell: Optimization by Integrated Artificial Neural Network and Particle Swarm Technique. *Heliyon*, 9(12).
33. Bendine, K., Pereira, J.L.J., Gomes, G.F., 2023. Energy Harvesting Enhancement of Nonuniform Functionally Graded Piezoelectric Beam Using Artificial Neural Networks and Lichtenberg Algorithm. *Structures*, 105271.
34. Yang, S., Tian, X., Zhang, Q., Jiang, J., Dong, P., Tan, J., Meng, Y., Liu, P., Bai, H., Song, J., 2023. Microorganism Inspired Hydrogels: Optimization by Response Surface Methodology and Genetic Algorithm Based on Artificial Neural Network. *European Polymer Journal*, 201, 112497.
35. Wang, C., He, Q., Li, Z., Yu, J., Bello, I.T., Zheng, K., Han, M., Ni, M., 2024. A Novel in-

- Tube Reformer for Solid Oxide Fuel Cell for Performance Improvement and Efficient Thermal Management: A Numerical Study Based on Artificial Neural Network and Genetic Algorithm. *Applied Energy*, 357, 122030.
36. Cinar, A.C., Natarajan, N., 2022. An Artificial Neural Network Optimized by Grey Wolf Optimizer for Prediction of Hourly Wind Speed in Tamil Nadu, India. *Intelligent Systems with Applications*, 16, 200138.
37. Bernard, J., Popescu, E., Graf, S., 2022. Improving Online Education through Automatic Learning Style Identification Using a Multi-Step Architecture with Ant Colony System and Artificial Neural Networks. *Applied Soft Computing*, 131, 109779.
38. Zhang, H., Nguyen, H., Bui, X.-N., Nguyen-Thoi, T., Bui, T.-T., Nguyen, N., Vu, D.-A., Mahesh, V., Moayedi, H., 2020. Developing a Novel Artificial Intelligence Model to Estimate the Capital Cost of Mining Projects Using Deep Neural Network-Based Ant Colony Optimization Algorithm. *Resources Policy*, 66, 101604.
39. Mirjalili, S., 2015. How Effective Is the Grey Wolf Optimizer in Training Multi-Layer Perceptrons. *Applied Intelligence*, 43, 150-161.
40. Gülcü, Ş., 2022. Training of the Feed Forward Artificial Neural Networks Using Dragonfly Algorithm. *Applied Soft Computing*, 124, 109023.
41. Turkoglu, B., Kaya, E., 2020. Training Multi-Layer Perceptron with Artificial Algae Algorithm. *Engineering Science Technology, an International Journal*, 23(6), 1342-1350.
42. Qaddoura, R., Faris, H., Aljarah, I., Castillo, P.A., 2021. Evocluster: An Open-Source Nature-Inspired Optimization Clustering Framework. *SN Computer Science*, 2, 1-12.
43. Ataseven, B., 2013. Yapay Sinir Ağları ile Öngörü Modellemesi. *Öneri Dergisi*, 10(39), 101-115.
44. Yang, X.-S., Deb, S., 2009. Cuckoo Search Via Lévy Flights. 2009 World Congress on Nature & Biologically inspired computing (NaBIC), Coimbatore, India, 210-214.
45. Pu, Y., Song, J., Wu, M., Xu, X., Wu, W., 2023. Node Location Using Cuckoo Search Algorithm with Grouping and Drift Strategy for Wsn. *Physical Communication*, 59, 102088.
46. Mirjalili, S., Mirjalili, S.M., Lewis, A., 2014. Grey Wolf Optimizer. *Advances in Engineering Software*, 69, 46-61.
47. Abualigah, L., Abd Elaziz, M., Sumari, P., Geem, Z.W., Gandomi, A.H., 2022. Reptile Search Algorithm (RSA): A Nature-Inspired Meta-Heuristic Optimizer. *Expert Systems with Applications*, 191, 116158.
48. Mirjalili, S., 2016. Sca: A Sine Cosine Algorithm for Solving Optimization Problems. *Knowledge-Based Systems*, 96, 120-133.