

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**BÜYÜK BOYUTLU VERİLER İÇİN METASEZGİSEL  
YÖNTEMLER İLE ÖZNİTELİK İNDİRGEMEDE YENİ BİR  
YAKLAŞIM GELİŞTİRİLMESİ**

**DOKTORA TEZİ**

**Esin Ayşe ZAIMOĞLU**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilgisayar Mühendisliği Bilim Dalı**

**HAZİRAN 2023**



T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**BÜYÜK BOYUTLU VERİLER İÇİN METASEZGİSEL  
YÖNTEMLER İLE ÖZNİTELİK İNDİRGEMEDE YENİ BİR  
YAKLAŞIM GELİŞTİRİLMESİ**

**DOKTORA TEZİ**

**Esin Ayşe ZAIMOĞLU**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Bilgisayar Mühendisliği Bilim Dalı**

**Tez Danışmanı: Doç.Dr.Nilüfer YURTAY**

**HAZİRAN 2023**



Esin Ayşe Zaimođlu tarafından hazırlanan “Büyük Boyutlu Veriler İçin Metasezgisel Yöntemler İle Öznitelik İndirgemedede Yeni Bir Yaklaşım Geliştirilmesi” adlı tez çalışması 09.06.2023 tarihinde aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Sakarya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Bilim Dalı’nda Doktora tezi olarak kabul edilmiştir.

### Tez Jürisi

**Jüri Başkanı :**            **Unvan Adı SOYADI** .....  
Sakarya Üniversitesi

**Jüri Üyesi :**            **Doç.Dr.Nilüfer YURTAY (Danışman)** .....  
Sakarya Üniversitesi

**Jüri Üyesi :**            **Unvan Adı SOYADI** .....  
Sakarya Üniversitesi

---

**Jüri Üyesi :**            **Unvan Adı SOYADI** .....  
**(Gerekliyse)**            ..... Üniversitesi

**Jüri Üyesi :**            **Unvan Adı SOYADI** .....  
**(Gerekliyse)**            ..... Üniversitesi



## **ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ**

Sakarya Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim-Öğretim Yönetmeliğine ve Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesine uygun olarak hazırlamış olduğum “BÜYÜK BOYUTLU VERİLER İÇİN METASEZGİSEL YÖNTEMLER İLE ÖZNİTELİK İNDİRGEMEDE YENİ BİR YAKLAŞIM GELİŞTİRİLMESİ” başlıklı tezin bana ait, özgün bir çalışma olduğunu; çalışmamın tüm aşamalarında yukarıda belirtilen yönetmelik ve yönergeye uygun davrandığımı, tezin içerdiği yenilik ve sonuçları başka bir yerden almadığımı, tezde kullandığım eserleri usulüne göre kaynak olarak gösterdiğimi, bu tezi başka bir bilim kuruluna akademik amaç ve unvan almak amacıyla vermediğimi ve 20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince Sakarya Üniversitesi’nin abonesi olduğu intihal yazılım programı kullanılarak Enstitü tarafından belirlenmiş ölçütlere uygun rapor alındığını, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun ortaya çıkması halinde doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi beyan ederim.

(...../...../20.....).

Esin Ayşe ZAIMOĞLU





*Kızım Ada'ya...*



## **TEŐEKKÜR**

Tez alıőmamın her aőamasında deęerli tecrübeleri ile bana yol gostererek desteklerini esirgemeyen danıőman hocam Do.Dr.Nilüfer YURTAY'a sonsuz teőekkürlerimi sunarım. Akademik hayatım boyunca beni destekleyen ok deęerli aileme ve kıymetli arkadaşlarıma da yürekten teőekkür ederim.

Esin Ayőe Zaimoęlu



## İÇİNDEKİLER

### Sayfa

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ .....	v
TEŞEKKÜR .....	ix
İÇİNDEKİLER .....	xi
KISALTMALAR .....	xiii
TABLO LİSTESİ .....	xv
ŞEKİL LİSTESİ .....	xvii
ÖZET.....	xix
SUMMARY .....	xxi
<b>1. GİRİŞ</b> .....	<b>1</b>
1.1. Problemin Tanımı.....	1
1.2. Tezin Amacı ve Önemi .....	2
1.3. Özgün Değer .....	5
1.4. Tez Organizasyonu.....	6
<b>2. KAYNAK ARAŞTIRMASI</b> .....	<b>7</b>
<b>3. MATERYAL VE YÖNTEM</b> .....	<b>21</b>
3.1. Kaos Teoremi .....	21
3.1.1. Kaotik haritalar .....	22
3.1.1.1. Logistic kaotik haritası.....	23
3.1.1.2. Piecewise kaotik haritası.....	24
3.1.1.3. Singer kaotik haritası .....	25
3.1.1.4. Sinusoidal kaotik haritası .....	26
3.1.1.5. Tent kaotik haritası.....	27
3.2. Sınıflandırma ve Kümeleme Algoritmaları.....	28
3.2.1. K en yakın komşu (k-NN) algoritması.....	29
3.2.1.1. Minkowski mesafe ölçümü .....	30
3.2.1.2. Öklid mesafe ölçümü .....	30
3.2.1.3. Manhattan mesafe ölçümü .....	31
3.2.2. Destek vektör makineleri (SVM).....	32
3.2.2.1. Doğrusal olarak ayrılabilen destek vektör makineleri .....	33
3.2.2.2. Doğrusal olarak ayrılamayan destek vektör makineleri.....	33
3.3. Öznitelik Seçimi .....	35
3.3.1. Filtreleme model .....	36
3.3.2. Sarmalayıcı model.....	37
3.3.3. Hibrid/Embedded model .....	38
3.4. Optimizasyon Algoritmaları.....	38
3.5. Metasezgisel Optimizasyon Algoritmaları .....	39
3.6. Doğa Esinli Optimizasyon Algoritmaları .....	40
3.6.1. Orijinal at sürüsü algoritması (HOA).....	41
3.6.1.1. Otlatma.....	44
3.6.1.2. Hiyerarşi.....	45
3.6.1.3. Sosyallik.....	46

3.6.1.4. Taklit .....	46
3.6.1.5. Savunma .....	47
3.6.1.6. Dolaşım .....	48
3.6.2. Öznitelik seçme problemi için ikili at sürüsü algoritması .....	50
3.6.2.1. İkili arama uzayında çözüm popülasyonun tasarımı .....	54
3.6.2.2. Sınıflandırıcı algoritması ve uygunluk değerlendirmesi .....	55
3.6.2.3. Atların arama stratejisi .....	57
3.6.2.4. Yerel arama tabanlı pozisyon güncelleme stratejisi .....	63
3.6.2.5. Sürü merkezi bulma fonksiyonu .....	63
3.6.2.6. Minimum mesafe fonksiyonu .....	64
3.6.2.7. Kaotik ikili HOA algoritması .....	65
3.6.2.8. BHOAFS algoritması temel kabulleri .....	68
3.6.2.9. Kullanılan veri setleri ve değerlendirme metrikleri .....	69
<b>4. ARAŞTIRMA BULGULARI VE TARTIŞMA .....</b>	<b>73</b>
4.1. BHOAFS'nin Performansı .....	73
4.2. BCHOAFS'nin Performansı .....	78
4.3. Geliştirilen Yöntemin Literatürdeki Algoritmalar ile Performans Kıyaslaması .....	83
4.4. Yakınsama Grafikleri .....	89
4.5. İstatistiksel Test Sonuçları .....	98
4.5.1. Friedman test sonuçları .....	98
4.5.2. Wilcoxon test sonuçları .....	102
<b>5. SONUÇ VE ÖNERİLER .....</b>	<b>105</b>
<b>KAYNAKLAR .....</b>	<b>107</b>
<b>ÖZGEÇMİŞ .....</b>	<b>115</b>

## KISALTMALAR

<b>ABC</b>	: Yapay Arı Kolonisi
<b>ABO</b>	: Yapay Kelebek Algoritması
<b>ACO</b>	: Karınca Kolonisi Optimizasyonu
<b>ALO</b>	: Karınca Aslanı Algoritması
<b>A<math>\beta</math>HC</b>	: Adaptif $\beta$ -tepe tırmanışı
<b>BA</b>	: Yarasa Algoritması
<b>BCHOAFS</b>	: Nitelik Seçimi İçin Kaotik İkili At Sürüsü Algoritması
<b>BDA</b>	: İkili Karahindiba Algoritması
<b>BHOAFS</b>	: Nitelik Seçimi İçin İkili At Sürüsü Algoritması
<b>BSO</b>	: Beyin Fırtınası Algoritması
<b>CRO</b>	: Kimyasal Reaksiyon Algoritması
<b>CSA</b>	: Karga Arama Algoritması
<b>D</b>	: Savunma
<b>DA</b>	: Yusufçuk Algoritması
<b>DE</b>	: Diferansiyel Evrim Algoritması
<b>DNN</b>	: Derin Sinir Ağlarını
<b>EA</b>	: Evrimsel Algoritma
<b>EO</b>	: Denge Optimizasyon Algoritması
<b>FA</b>	: Ateş Böceği Algoritması
<b>FLA</b>	: Kurbağa Sıçraması Algoritması
<b>FOA</b>	: Orman Optimizasyon algoritmasına
<b>FS</b>	: Öznitelik seçimi
<b>FSA</b>	: Balık Sürü Algoritması
<b>G</b>	: Otlatma
<b>GA</b>	: Genetik Algoritma
<b>GOA</b>	: Grasshopper Algoritması
<b>GSA</b>	: Yerçekimi Arama Algoritması

<b>GSO</b>	: Grup Arama Optimizasyon Algoritması
<b>H</b>	: Hiyerarşi
<b>HHO</b>	: Harris Hawk Optimizasyon Algoritması
<b>HOA</b>	: At Sürüsü Algoritması
<b>KKO</b>	: Karınca Kolonisi Optimizasyonu
<b>k-NN</b>	: k En Yakın Komşu
<b>LDA</b>	: Doğrusal Diskriminant Analizi
<b>MFO</b>	: Ay Alevi Optimizasyonu
<b>PCA</b>	: Temel Bileşen Analizi
<b>PSO</b>	: Parçacık Sürü Algoritması
<b>R</b>	: Dolaşım
<b>S</b>	: Sosyallik
<b>SA</b>	: Tavlama Benzetimi Algoritması
<b>SCA</b>	: Sinüs Kosinüs Algoritması
<b>SFO</b>	: Yelken Balığı Algoritması
<b>SMA</b>	: Sümüksü Kalıp Algoritması
<b>SSA</b>	: Salp Sürü Algoritması
<b>SVM</b>	: Destek Vektör Makinesi
<b>SZ</b>	: Süzü Zekası
<b>T</b>	: Taklit
<b>WOA</b>	: Balina Optimizasyon Algoritması
<b>YSA</b>	: Yapay Sinir Ağı



## TABLO LİSTESİ

### Sayfa

<b>Tablo 3.1.</b> Seçilen kaotik haritalara ait matematiksel gösterim.....	23
<b>Tablo 3.2.</b> Atların arama stratejileri genel bakış. ....	61
<b>Tablo 3.3.</b> Kullanılan veri setleri.....	69
<b>Tablo 4.1.</b> BHOAFS'nin k-NN ve SVM sınıflandırıcılarına ait doğruluk ve öznitelik karşılaştırması. ....	76
<b>Tablo 4.2.</b> BHOAFS'nin sınıflandırıcılarına ait iyi bilinen metasezgisel algoritmalar ile ortalama uygunluk fonksiyonu karşılaştırması. ....	76
<b>Tablo 4.3.</b> BHOAFS k-NN sınıflandırıcısına ait kaotik varyasyonların doğruluk değerleri.....	80
<b>Tablo 4.4.</b> BHOAFS k-NN sınıflandırıcısına ait kaotik varyasyonların öznitelik sayıları.....	81
<b>Tablo 4.5.</b> BHOAFS k-NN sınıflandırıcısına ait kaotik varyasyonların ortalama doğruluk ve öznitelik sayısı. ....	82
<b>Tablo 4.6.</b> BHOAFS'nin iyi bilinen optimizasyon algoritmaları ile ortalama doğruluk karşılaştırması. ....	85
<b>Tablo 4.7.</b> BHOAFS'nin iyi bilinen ikili optimizasyon algoritmaları ile ortalama doğruluk karşılaştırması. ....	86
<b>Tablo 4.8.</b> BHOAFS'nin iyi bilinen ikili optimizasyon algoritmaları ile öznitelik sayısı karşılaştırması. ....	87
<b>Tablo 4.9.</b> BHOAFS'nin iyi bilinen ikili optimizasyon algoritmalarıyla CPU süresi (saniye) karşılaştırılması. ....	88
<b>Tablo 4.10.</b> Friedman test sonuçları. ....	99
<b>Tablo 4.11.</b> Önerilen algoritmalara ait friedman istatistiksel test sonuçları.....	100
<b>Tablo 4.12.</b> Önerilen algoritmalara ait friedman ortalama rank değeri.....	101
<b>Tablo 4.13.</b> Önerilen algoritmaların wilcoxon istatistiksel testine ait p değeri (anlamlılık düzeyi 0.05). ....	103
<b>Tablo 4.14.</b> Önerilen algoritmaya ait wilcoxon istatistiksel test sonuçları.....	104



## ŞEKİL LİSTESİ

### Sayfa

Şekil 3.1. Logistic kaotik haritası yayılım spektrumu. ....	24
Şekil 3.2. Piecewise kaotik haritası yayılım spektrumu. ....	25
Şekil 3.3. Singer kaotik haritası yayılım spektrumu. ....	26
Şekil 3.4. Sinusoidal kaotik haritası yayılım spektrumu. ....	27
Şekil 3.5. Tent kaotik haritası yayılım spektrumu. ....	28
Şekil 3.6. k değeri seçimi. ....	31
Şekil 3.7. Doğrusal olarak ayrılabilen veri setleri için hiper-düzlemin belirlenmesi	33
Şekil 3.8. Doğrusal olarak ayrılamayan veri seti(Kavzoğlu & Çölkesen, 2010). ....	34
Şekil 3.9. Doğrusal olarak ayrılamayan verilerin hiper-düzlemin belirlenmesi (Kavzoğlu & Çölkesen, 2010). ....	34
Şekil 3.10. Kernel fonksiyonu ile verinin daha yüksek boyuta dönüştürülmesi. ....	35
Şekil 3.11. Orjinal HOA sözde kodu. ....	42
Şekil 3.12. Orjinal HOA'nın tüm çalışma adımları. ....	43
Şekil 3.13. Atların otlama davranışı. ....	44
Şekil 3.14. Atların hiyerarşi davranışı. ....	45
Şekil 3.15. Atların sosyalleşme davranışı. ....	46
Şekil 3.16. Atların taklit etme davranışı. ....	47
Şekil 3.17. Atların savunma davranışı. ....	48
Şekil 3.18. Atların dolaşma davranışı. ....	49
Şekil 3.19. HOA algoritmasının sıralama süreci. ....	50
Şekil 3.20. Önerilen BHOAFS için akış şeması. ....	53
Şekil 3.21. Temsili bir ikili çözüm vektörü (Turabieh, 2019). ....	55
Şekil 3.22. $\alpha$ atları için arama stratejisi. ....	57
Şekil 3.23. $\beta$ atları için arama stratejisi. ....	58
Şekil 3.24. $\gamma$ atları için arama stratejisi. ....	59
Şekil 3.25. $\delta$ atları için arama stratejisi. ....	60
Şekil 3.26. Minimum mesafe fonksiyonu. ....	65
Şekil 3.27. BCHOAFS sözde kodu. ....	67
Şekil 4.1. Vehicle veri seti için yakınsama grafiği. ....	89
Şekil 4.2. Zoo veri seti için yakınsama grafiği. ....	90
Şekil 4.3. Sonar veri seti için yakınsama grafiği. ....	90
Şekil 4.4. BreastEW veri seti için yakınsama grafiği. ....	91
Şekil 4.5. Arrhythmia veri seti için yakınsama grafiği. ....	91
Şekil 4.6. Hillvalley veri seti için yakınsama grafiği. ....	92
Şekil 4.7. Hepatitis veri seti için yakınsama grafiği. ....	92
Şekil 4.8. CNAE veri seti için yakınsama grafiği. ....	93
Şekil 4.9. Wine veri seti için yakınsama grafiği. ....	93
Şekil 4.10. Breast Cancer veri seti için yakınsama grafiği. ....	94
Şekil 4.11. Ionosphere veri seti için yakınsama grafiği. ....	94
Şekil 4.12. Lung Cancer veri seti için yakınsama grafiği. ....	95
Şekil 4.13. Heart veri seti için yakınsama grafiği. ....	95

<b>Şekil 4.14.</b> Movementlibras veri seti için yakınsama grafiği.....	96
<b>Şekil 4.15.</b> Waveform veri seti için yakınsama grafiği.....	96
<b>Şekil 4.16.</b> Soybean Small veri seti için yakınsama grafiği.....	97
<b>Şekil 4.17.</b> Clean veri seti için yakınsama grafiği.....	97
<b>Şekil 4.18.</b> Semeion veri seti için yakınsama grafiği.....	98

# BÜYÜK BOYUTLU VERİLER İÇİN METASEZGİSEL YÖNTEMLER İLE ÖZNETELİK İNDİRGEMEDE YENİ BİR YAKLAŞIM GELİŞTİRİLMESİ

## ÖZET

Teknolojinin gelişmesi ve çevrimiçi dünyanın genişlemesi ile iş dünyası, bilim, sağlık ve mühendislik gibi bir çok alanda çok büyük boyut ve tür içeren gerçek dünya verileri ortaya çıkmıştır. Günümüz dünyasında bu verilerin en faydalı kullanımı ve en az saklama maliyeti hedeflendiğinden, ancak belirli anlam taşıyan verilerin saklanması gerekmektedir. Boyutları büyüyen ham verideki örüntüler bulunmadığı sürece veriler gereksizdir. Büyük boyutlu veriyi işlemek ise onları kullanan sistemlerin verimliliğini doğrudan etkileyen bir darboğaz problemidir. Büyük veri ve büyük boyutlu veri için nitelik azaltma olarak da adlandırılan öznetelik seçme problemi, orijinal özelliklerin ayırt edici gücünü koruyarak veri boyutunu indirgemeyi amaçlamaktadır. Öznetelik seçimi, öğrenme sürecinin önemli adımlarından biridir. En iyi çözümün büyük bir arama uzayında uygun sürede bulunmasını gerektirir. Öznetelik seçim problemi için tasarlanan algoritmalar, orijinal özelliklerin hesapsal maliyetini azaltırken en uygun özellik alt kümesini bulma amaçlı rastgele veya sezgisel arama stratejileri kullanır. Veri hacmi arttıkça veri ile yapılacak tüm işlemler de doğrusal olarak zorlaşmaktadır. Bu sebeple ortaya atılan öznetelik seçim probleminin çözümü için literatürde birden çok yöntem kullanılmıştır.

Klasik sezgisel algoritmalar büyük boyutlu ve karmaşık problemlere uygulandığında tasarımındaki avantaj ve etkinliklerini kaybetmektedirler. Çünkü bu algoritmalar, yerel çözümlere takılma, kararsız performans sergileme, bellek verimliliğinden yoksun olma gibi kısıtlamalara sahip olabilirler (Qiu & Liu, 2016). Bu nedenle bu tip problemlerin çözümü için daha etkili algoritmalara ihtiyaç duyulmaktadır. Öznetelik seçim probleminin çözümü için etkili yaklaşımlardan biri de Metasezgisel algoritmalarlardır. Doğadan esinlemeli algoritmalar çözüm uzayında, sezgisel bir arama yaparak daha makul sürelerde daha uygun sonuçların elde edilmesini sağlar. Bu kapsamda, ana veri kümesinin küçük ama etkili bir alt kümesi olarak tespit edilecek olan niteliklerin belirlenmesi için algoritmanın küresel arama yeteneğini ve sezgiselliğini kaybetmeden, bir yerel arama fonksiyonu ile desteklenmesinin başarıyı artıracığı düşünülmüştür (X.-M. Hu vd., 2017).

Sezgisel algoritmalar evrimsel algoritmalar ve sürü temelli algoritmalar temel iki grup olarak nitelendirilir. Evrimsel algoritmalar ile gerçek dünya problemlerine başarılı sonuçlar üretilebilmektedir. Bu algoritmalar Genetik Algoritma, Evrimsel Strateji ve Diferansiyel Evrim algoritması en bilinen örneklerdendir. Doğada var olan çeşitli canlıların davranışlarını taklit ederek geliştirilen sürü temelli algoritmalar ise gün geçtikçe popülerliği artan algoritmalarlardır. Bu algoritmalar Parçacık Sürü Optimizasyon Algoritması, Yarasa Algoritması, Karınca Kolonisi Optimizasyon Algoritması, Yapay Arı Kolonisi Algoritması örnek verilebilir.

At Sürüsü Algoritması (HOA) ise, en yeni sürü tabanlı optimizasyon algoritmalarından biridir. HOA, doğada birlikte yaşayan atların sürü olarak davranışlarından ilham alarak tasarlanmıştır. Atların farklı yaşlarda gösterdikleri otlama, hiyerarşi, sosyallik, taklit, savunma ve dolaşma olmak üzere altı kategorideki sosyal davranışları algoritmanın temelini oluşturmaktadır. Literatürde optimizasyon algoritmalarının ikili arama uzayına taşınarak sıkça öznitelik seçim problemine uygulandığı gözlenmektedir.

Bu tez kapsamında öznitelik seçimi için HOA'nın ikili bir versiyonu iyi bilinen iki farklı sınıflandırma algoritması (k-NN ve SVM) ile önerilmiştir. HOA'nın nitelik seçimi için önerilen ikili versiyonu öncelikle küçük boyutlu (10,20,30) veriler ile test edilmiştir. Ardından büyük boyutlu verilerde de efektif sonuçlar üretmesi amacıyla yerel arama tekniği olan Minimum Mesafe Fonksiyonu ve Sürü Merkezi Bulma Fonksiyonu geliştirilerek İkili At Sürüsü Algoritması (BHOAFS) önerilmiştir. BHOAFS'nin eklenen yöntemler ile yerel ve küresel arama yeteneği artmıştır. BHOAFS'nin düşük, orta ve yüksek boyutlu (10,20,30,100,500 ve 1000) problemlerde başarısı test edilmiştir. BHOAFS-k-NN algoritmasının BHOAFS-SVM'ye göre daha iyi sonuçlar ürettiği gözlemlenmiştir.

Literatürde metasezgisel optimizasyon algoritmaların keşif ve sömürü sürecindeki dengeyi sağlayarak süreci güçlendirmek için kaos teorisi, levy uçuşu, zeki arama, kuantum davranışı gibi çeşitli stratejiler kullanılmıştır. Bu tez çalışmasında HOA'nın önerilen ikili versiyonu algoritmanın başarısı ve kararlılığını arttırmak amacıyla iyi bilinen beş farklı kaotik harita ile güçlendirilerek Kaotik İkili At Sürüsü Algoritması BCHOAFS olarak adlandırılan yeni bir yaklaşım geliştirilmiştir. Önerilen BCHOAFS, büyük ölçekli veriler için özel olarak tasarlanmış bir optimizasyon algoritması olan HOA'nın özellik seçim problemleri için özel olarak geliştirilmiş ilk ikili kaotik tabanlı algoritmadır. k-NN sınıflandırmasını kullanan BHOAFS-k-NN yöntemi daha iyi sonuçlar ürettiğinden, beş kaotik harita ile birleştirilmiş ve BCHOAFS-Logistic, BCHOAFS-Tent, BCHOAFS-Piecewise, BCHOAFS-Singer, BCHOAFS-Sinusoidal olarak adlandırılmıştır. Geliştirilen bu yaklaşımın performansı farklı boyuttaki on sekiz veri seti ile ölçülmüştür. Kaotik tabanlı yeni yaklaşım literatürde öznitelik seçimi için geliştirilen çeşitli algoritmalar ile kıyaslanmış, doğruluk, seçilen öznitelik sayısı, uygunluk fonksiyonu testleri yapılarak güvenilirliği sağlanmıştır. Önerilen yaklaşımın kıyaslanan algoritmalarından daha performanslı olduğu istatistiksel yöntemlerden Friedman işaret testi ve post hoc Wilcoxon testi kullanılarak doğrulanmıştır.

## **DEVELOPING A NEW APPROACH TO FEATURE SELECTION WITH METAHEURISTIC METHODS FOR LARGE SCALE DATA**

### **SUMMARY**

With the development of technology and the expansion of the online world, real-world data of enormous size and type has emerged in many fields such as business, science, health and engineering. In today's world, since the most beneficial use of this data and the least storage cost is aimed, only data that has a certain meaning should be stored. Data is useless unless there are patterns in the growing raw data. Processing large-scale data is a bottleneck problem that directly affects the efficiency of the systems that use them. Feature selection problem, also called feature reduction for big data and large size data, aims to reduce the data size while preserving the distinctive power of the original features. Feature selection is one of the important steps of the learning process. It requires the best solution to be found in a large search space in a reasonable amount of time. Algorithms designed for the feature selection problem use random or heuristic search strategies to find the optimal subset of features while reducing the computational cost of the original features. As the volume of data increases, all operations with the data become linearly difficult. For this reason, more than one method has been used in the literature to solve the proposed feature selection problem.

When classical heuristic algorithms are applied to large and complex problems, they lose their advantages and effectiveness in design. Because these algorithms may have limitations such as being stuck in native solutions, unstable performance, and lack of memory efficiency. Therefore, more effective algorithms are needed to solve such problems. Metaheuristic algorithms are one of the effective approaches for solving the feature selection problem. Nature-inspired algorithms perform an intuitive search in the solution space, resulting in more appropriate results in more reasonable time. In this context, it is thought that supporting the algorithm with a local search function without losing its global search capability and heuristics in order to determine the features that will be detected as a small but effective subset of the main dataset will increase the success.

Heuristic algorithms, evolutionary algorithms and swarm-based algorithms are characterized as the two main groups. Successful results can be produced for real world problems with evolutionary algorithms. Among these algorithms, Genetic Algorithm, Evolutionary Strategy and Differential Evolution algorithm are the best-known examples. Swarm-based algorithms, which are developed by imitating the behavior of various living things in nature, are the algorithms that are increasing in popularity day by day. Particle Swarm Optimization Algorithm, Bat Algorithm, Ant Colony Optimization Algorithm, Artificial Bee Colony Algorithm can be given as examples to these algorithms.

Horse Herd Algorithm (HOA) is one of the newest herd-based optimization algorithms. HOA is a new metaheuristic algorithm created by modeling the herd behavior of horses and was designed with inspiration from the herd behavior of horses living together in nature developed for large scale optimization problems.

The social behaviors of horses in six categories: grazing, hierarchy, sociability, imitation, defense and wandering at different ages form the basis of the algorithm. In the literature, it is observed that optimization algorithms are frequently applied to the feature selection problem by moving to binary search space.

One of the most challenging and common problems in machine learning is the Feature Selection (FS) process, which reduces the dataset size by finding optimal subsets of features. This thesis proposes the binary version of HOA, which mimics the life cycles and searching behaviors of horses, has been applied to a wrapper-based FS problem using classification algorithms (BHOAFS-kNN and BHOAFS-SVM). A binary version of HOA is proposed for feature selection with two well-known classification algorithms (k-NN and SVM).

In this thesis, a binary version of HOA is proposed for feature selection with two well-known classification algorithms (k-NN and SVM). The proposed binary version of the HOA for feature selection was first tested with small data. Then, in order to produce effective results in large-scale data, the Minimum Distance Function and Herd Center Finding Function, which are local search techniques, were developed and Binary Horse Herd Algorithm (BHOAFS) was proposed. Local and global search capability of BHOAFS has increased with the added methods. The success of BHOAFS has been tested in low, medium and high dimensional (e.g., 10, 20, 30, 100, 500 and 1000) problems. It has been observed that the BHOAFS-k-NN algorithm produces better results than BHOAFS-SVM.

In the literature, various strategies such as chaos theory, levy flight, intelligent search, and quantum behavior have been used to strengthen the process by providing the balance in the discovery and exploitation process of metaheuristic optimization algorithms. This paper proposes the binary version of the HOA as a wrapper FS method to solve the FS problem. The proposed algorithm is a binary chaotic horse herd optimization algorithm for feature selection (BCHOAFS).

This thesis proposes the binary version of HOA, which mimics the life cycles and searching behaviors of horses, has been applied to a wrapper-based FS problem using classification algorithms (BHOAFS-kNN and BHOAFS-SVM). In this thesis, a new approach called Binary Chaotic Horse Herd Algorithm BCHOAFS has been developed by strengthening the proposed binary version of HOA with five different well-known chaotic maps in order to increase the success and stability of the algorithm.

The proposed BCHOAFS is the first binary chaotic based algorithm specially developed for feature selection problems of HOA, an optimization algorithm specially designed for large-scale data. Since the BHOAFS-k-NN method using k-NN classification produced better results, it was combined with five chaotic maps and named as BHOAFS-Logistics (BCHOAFS1), BCHOAFS - Tent (BCHOAFS2), BCHOAFS - Piecewise (BCHOAFS3), BCHOAFS - Singer (BCHOAFS4), and BCHOAFS - Sinusoidal (BCHOAFS5). The Age Determination process used in the proposed algorithm allows the horses to do a global search in the search space. The SMF operator, which is proposed as a local search technique, has been developed to overcome the disadvantages, such as early convergence while preserving population diversity. A comprehensive study was conducted using 18 standard datasets from the UCI repository of varying sizes and characters to evaluate the effectiveness of the proposed BHOAFS and BCHOAFS versions.

The chaotic-based new approach has been compared with various popular algorithms developed for feature selection in the literature, and its reliability has been ensured by



performing tests in terms of accuracy, number of selected features, fitness function and runtime. It has been verified by using Friedman Signed Rank and post hoc Wilcoxon test from statistical methods that the proposed approach is more performant than the compared algorithms.

The performance of the proposed methods was compared with various methods such as well-known optimization algorithms in the literature such as GA, PSO, ALO, GWO, SSA, and binary optimization algorithms such as BGA, BPSO, BALO, BGWO, and BSSA. When the FS problem is considered a multi-objective problem, the classification accuracy should be at the highest level, and the number of selected features should be at the lowest level. When the results were examined, the best performance among the recommended algorithms was found to belong to BCHOAFS3 and BCHOAFS4 algorithms (BCHOAFS-Piecewise and BCHOAFS-Singer) in terms of classification accuracy and FS.

In general, applying chaotic maps to optimization algorithms is easy and practical. According to this study, chaotic maps can improve results when applied to previously proposed algorithms. The main reason for this improvement is that the algorithm can more accurately since the initial conditions are designed in advance. Thus, bit-changing operations during the iteration will result in more specific exploitation and exploration and a balance will be established between the algorithm's local and global search operations. Taking full advantage of this situation, our proposed algorithm proves that chaotic maps are one of the best enhancement options for currently proposed algorithms when the results obtained are examined. Based on the promising results of BCHOAFS, there is enough evidence to compare its performance with similar methods. Finally, the results show that the proposed BCHOAFS achieves high competitive results and can be expressed as a multi-objective optimization problem. The most important result of this study is that BCHOAFS versions are useful selection algorithms. When the SMF operator designed as a local search strategy and chaotic maps are combined, the results prove the novelty of the proposed algorithm.

In the proposed method in future studies, the movement of the horse herd directed towards the center of the search space with the SMF function can be improved with a different solution in terms of exploration capability by calculating the movement sensitivity. HOA can be combined with other algorithms to improve the multi-purpose version. Customized solutions for the position of each horse in the search space can be developed by supporting various statistical methods. The U-shaped transfer function used in the transition from continuous space to discrete space can be tested by replacing it with other transfer functions. The classifiers used for feature selection performance measurement can be replaced with other classification algorithms and their accuracy can be compared.



## 1. GİRİŞ

Günümüz dünyasında teknolojinin gelişmesi ile bilgisayar performanslarındaki hızlı ilerleme, donanımların ucuzlaması ve veri tabanlarındaki verilerin artması ile iş dünyası, bilim ve mühendislik alanında çok büyük boyutlu çeşitli gerçek dünya verileri ortaya çıkmıştır. Bu veriler ham veri olarak nitelendirilir. Ham verilerin maximum kullanım ve minimum saklama maliyeti ile, işe yarar ve anlamlı veriler oluşturularak saklanması hedeflenmektedir. Büyük veri senaryoları ise milyonlarca satır / örnek ve yüz binlerce özellikten oluşmaktadır. Bu veriler ile ilgili işlem yapılınca kadar sürekli olarak saklanırken, ancak sınırlı sayıda özellikten faydalı bilgiler çıkarılabilmekte, makine öğrenimi ve veri madenciliği gibi alanlarda kullanılmaktadır. Boyutları büyüyen ham verideki örüntüler bulunmadığı sürece veriler gereksiz olarak nitelendirilmektedir. Böyle bir senaryoda elde edilen büyük boyutlu veriden umut verici eğitim verilerinin elde edilmesi çok önemli ancak zaman alıcıdır. Bu sebeple büyük boyutlu veri, veri kullanarak tasarlanan sistemlerin verimliliğini doğrudan etkileyen bir darboğaz problemidir. Büyük veri ve büyük boyutlu veri için öznelik azaltma olarak da adlandırılan özellik seçimi sorunu, orijinal özelliklerin ayırt edici gücünü korumayı amaçlamaktadır (Thuy & Wongthanavas, 2020). Büyük boyutlu veride, ayırt ediciliği yüksek bir nitelik alt kümesi belirlemek, tüm olası nitelik alt kümelerinin test edilmesini gerektirmektedir. Bu nedenle ilgili işlem hem karmaşıklık hem de zaman maliyeti açısından oldukça zordur. Büyük boyutlu veriden faydalı bilgi çıkarma, veriyi depolama ve işlem maliyeti sorunlarının çözümü de bu noktada önem kazanmıştır. Tüm bu nedenlerle büyük boyutlu veriler için var olanların ötesinde etkili ve verimli yeni teknikler geliştirilmesine ihtiyaç vardır.

### 1.1. Problemin Tanımı

Yapay zekanın geniş bir alt alanı olan makine öğrenmesi, bilgisayar öğrenmesini sağlayan algoritmalar ve teknikler tasarlar. Makine öğrenmesi uygulamalarının temel mantığı, çeşitli yöntemlerle veriden bağımsız olarak bilgi çıkarımı yapmak ve ortaya çıkan bu bilgiden yararlanmaktır. Dolayısıyla, makine öğrenmesi yöntemleri veri madenciliği ve istatistikle ilişkili olduğu kadar teorik bilgisayar bilimleriyle de temel

düzyeyde ilişkilidir. Makine öğrenmesi, çeşitli hesaplama yöntemleri ve belirli öğrenme mekanizmaları ile bilginin veriden elde edilmesini sağlayarak, sosyal hayata entegrasyonunu tasarlayan sistemin tamamını ifade etmektedir. Veri hacmi arttıkça veri ile yapılacak tüm işlemler de doğrusal olarak zorlaşmaktadır. Bu sebeple makine öğrenmesi algoritmalarında kullanılacak olan verinin boyuttan bağımsız olarak doğru ve etkin şekilde çalışması önemlidir. Öğrenmenin doğru şekilde yapılabilmesi için büyük boyutlu veriden aynı niteliğe sahip ancak daha küçük boyutlu veri elde edilerek makine öğrenmesi algoritmalarında kullanılması gerekmektedir.

Büyük boyutlu veride, ayırt ediciliği yüksek bir nitelik alt kümesi belirlemek, tüm olası nitelik alt kümelerinin test edilmesini gerektirmektedir. Bu nedenle ilgili işlem hem karmaşıklık hem de zaman maliyeti açısından oldukça zordur. Büyük boyutlu veriden faydalı bilgi çıkarma, veriyi depolama ve işlem maliyeti sorunlarının çözümü de bu noktada önem kazanmıştır. Büyük veri senaryoları ise milyonlarca satır / örnek ve yüz binlerce özellikten oluşmaktadır. Makine öğrenmesi işlemlerinde böyle bir senaryoda umut verici eğitim verilerinin elde edilmesi çok önemli ve zaman alıcıdır. Büyük boyutlu veri, tasarlanan analizlerin verimliliğini doğrudan etkileyen bir darboğaz problemidir. Bu amaçla ortaya atılan öznelik seçim probleminin çözümü için literatürde birden çok yöntem kullanılmıştır. Bu yöntemlerden biri de Metasezgisel algoritmaları kullanarak öznelik seçimi yapmaktır. Klasik sezgisel algoritmalar büyük boyutlu ve karmaşık problemlere uygulandığında tasarımındaki avantaj ve etkinliklerini kaybetmektedirler. Çünkü bu algoritmalar, yerel çözümlere takılma, kararsız performans sergileme, bellek verimliliğinden yoksun olma gibi kısıtlamalara sahip olabilirler (Qiu & Liu, 2016). Bu nedenle bu tip problemlerin çözümü için daha etkili algoritmalara ihtiyaç duyulmaktadır. Öznelik seçimi olarak da bilinen nitelik altkümesi seçimi, özellik indirgeme mühendisliğinin bir parçasıdır. Aynı sonucu verecek olan özneliklerin mümkün olan en küçük alt kümesinin tespitinin yapılarak makine öğrenmesi veya veri madenciliği süreçlerinde kullanılması olarak özetlenebilir.

## **1.2. Tezin Amacı ve Önemi**

Yapay zeka tekniklerini kullanarak veri ön işleme çalışmalarından olan özellik seçimi diğer bir deyişle nitelik indirgeme son yıllarda dikkat çekici hale gelmiştir (Brezočnik vd., 2018). Nitelik seçimi, büyük bir arama uzayında en iyi çözümü bulmayı

gerektirerek, öğrenme sürecine katkı sağlar. Özellik seçimindeki temel kural, orijinal özellikleri temsil ederken veriden gereksiz nitelikleri kaldırarak yüksek doğruluk sağlamaktır. Veriler arasındaki ilişkiyi herhangi bir uzman olmadan öğrenmek ve gereksiz verileri kaldırmak özellikle büyük hacimli veriler için kritik öneme sahiptir. Geliştirilen yapay zekâ süreçlerinde büyük boyutlu veriler ile çalışılırken benzer ve daha iyi performans elde edebilmek için veriyi doğru temsil eden nitelik alt gruplarının belirlenmesi önem kazanmaktadır. Nitelik seçiminin; ilgisiz, gereksiz veya gürültülü veriyi yok etme, öğrenme algoritmasının performansını geliştirme, hesaplama maliyetini azaltma, veri kümelerinin daha iyi anlaşılmasını sağlama, nitelik uzayının boyutunu azaltma ve depolama yeri gereksinimlerini azaltma gibi avantajları vardır. Bu sayede geleneksel makine öğrenme ve örüntü tanıma sistemleri, veri madenciliği süreçleri gibi yapay zeka teknikleri genellikle elde edilen bu küçük boyutlu veri kümelerinde daha iyi ve performanslı çalışmaktadır (Nguyen vd., 2020). Özellik seçimi, bir özellik alt kümesi seçerek veya ilgisiz olanları kaldırarak büyük boyutlu verileri işlemedeki darboğazın üstesinden gelmek için makine öğrenimi, veri madenciliği ve örüntü tanıma alanında yaygın olarak kullanılan tekniklerdendir (Anaraki vd., 2018).

Son yıllarda, Metasezgisel Algoritmaların öznelik seçim problemine uygulanması ile ilgili yapılan araştırmalar, hem teorik hem de pratik çalışmalar da dâhil olmak üzere, oldukça artmıştır. Örneğin Sürü Zekâsı Algoritmaları öznelik seçim problemine uygulanarak giderek artan bir uygulama alanı haline gelmiştir (Brezočnik vd., 2018). Sürü Zekâsı (SZ), doğadaki canlıların birlikte hareket edişlerini temel alan bir zeka türü olarak nitelendirilmektedir. Bu sebeple de sürü zekâsı kullanılan yöntemler son yıllarda araştırma popülerliği kazanmıştır. Bu yöntemlerin temel amacı ise canlıların davranış ve çevre etkileşimlerini baz alarak modellemektir. Metasezgisel algoritmaların anlaşılabilirliği ve uygulama kolaylığı olması dolayısı nedeni ile tüm bu yöntemler çeşitli problemlerin çözümünde sıklıkla kullanılmaya başlanmıştır. Canlıların davranış ve hareket modeli karmaşık optimizasyon problemlerini çözmek için yarar sağlayabilir (Talbi, 2009). Canlıların evrimi, metallerde soğutma işlemi, bir kolonideki karıncaların yaşam döngüsü, kuşların göçü, bağışıklık sistemi vb. konular metasezgisel algoritmaların temelini oluşturan doğa esinli algoritmalara ilham olmuştur. Doğadan esinli algoritmalar çözüm uzayında, sezgisel bir arama yaparak daha makul sürelerde daha uygun sonuçların elde edilmesini sağlar. Bu kapsamda, ana

veri kümesinin küçük ama etkili bir alt kümesi olarak tespit edilecek olan niteliklerin belirlenmesi için algoritmanın küresel arama yeteneği ve sezgiselliğini kaybetmeden, bir yerel arama fonksiyonu ile desteklenmesinin başarıyı artıracakı düşünölmüştür (X.-M. Hu vd., 2017).

Bu tez kapsamında; öznitelik seçim problemi için önerilen mevcut çözüm teknikleri incelenerek analiz edilmiş, mevcut algoritmalar amaca yönelik değıştirilerek daha verimli hale getirilmiş ve yeni bir çözüm önerisi geliştirilmiştir. Ayrıca, daha az sayıdaki veri boyutunda çalıştığında başarı gösteren algoritmanın büyük boyutlu veri boyutunda da aynı öğrenme stratejisiyle başarılı olamayabileceği düşünöldüğünden çeşitli yöntemlerle arama fonksiyonu desteklenip, stratejisi güçlendirilerek nitelik altkümesi seçiminde başarının arttırılacağı ıspatlanmıştır.

Tez çalışması, bir arama uzayında optimal çözüm bulmak amaçlı kullanılan mevcut metasezgisel optimizasyon algoritmasının, geliştirilerek ikili bir versiyonu ile büyük boyutlu veride, efektif bir şekilde ayırt edici nitelik alt gruplarını bulması amaçlanmıştır. Çalışma kapsamında aşağıdaki durumlar gerçekleştirilmiştir:

- Tez çalışmasına temel oluşturan yaklaşımda, özellik indirgeme işlemi için önerilen algoritma ile minimum bir alt küme bulunmuş ve doğrulanmıştır.
- Etkileşimli nitelik alt gruplarını belirleyebilmek için geliştirilen ikili optimizasyon yaklaşımının sezgisellik özelliğini kaybolmadan yerel ve küresel arama yeteneğinin çeşitli yöntemlerle arttırılması başarıyı arttırmıştır.
- Yaklaşımın boyut artışına bağılı olarak arama uzayının katlanarak büyümesi nedeniyle arama uzayını etkin ve verimli bir şekilde keşfedebilmesi, büyük arama uzayları için de mevcut çalışmalardan daha iyi ve verimli sonuçlar vermesi hedeflenmiş ve hedefe ulaşılmıştır.
- Algoritmanın verimliliğini göstermek için bazı standart veri setlerine dayanan sayısal ve istatistiksel deneyler yapılmıştır.
- Sonuçlar, algoritmanın, bilgi kaybı olmadan niteliklerin minimal bir alt kümesini bulmak için etkili bir araç olduğunu göstermiştir.

Bu tez kapsamında yapılan çalışmalar ile nitelik indirgeme probleminin daha hızlı, daha etkili ya da daha verimli çözölməsi amaçlanmış, mevcut teknikler incelenip eksikler tespit edilerek geliştirmek için yapılabilecekler analiz edilmiş, bu verilerin kılavuzluğunda yeni öneriler sunulurak çözüm sürecine katkı sağlanmışır. Burada

katkıdan kasıt, hem zamandan kazanç hem de mevcut verileri en etkili şekilde temsil edebilen bir çözüme ulaşmaktır.

### 1.3. Özgün Değer

Literatürde optimizasyon algoritmaları ve öznelik seçim problemleri üzerine birçok çalışma bulunmaktadır. Ancak, büyük verilerle özellik seçimi bir NP-Hard problemidir. Bu nedenle, tez çalışmasının temel amacı, büyük veriler üzerinde dahi etkili bir şekilde çalışan bir ikili özellik seçim algoritması geliştirmektir.

2020 yılında geliştirilen At Sürüsü Algoritması (HOA) (MiarNaeimi vd., 2021a), sürekli uzaydaki problemlere çözüm bulabilmek için önerilen ve yüksek boyutlu optimizasyon problemleri için doğadan ilham alan bir algoritmadır. HOA at sürülerinin yaşamını modellemektedir. Algoritma özellikle büyük boyutlu sürekli uzayda birçok probleme uygulanmış ve efektif sonuçlar alınmıştır.

Tez çalışmasında önerilen algoritma, büyük boyutlu verilerde hızlı ve etkili sonuçlar üreten At Sürüsü Algoritmasına dayalı olarak, özellik seçim problemine farklı bir bakış açısıyla geliştirilmiştir. Algoritmanın büyük veride başarılı olan fonksiyonları göz önünde bulundurularak ikili versiyonu nitelik seçim problemi için iyi bilinen sınıflandırıcılarla kodlanmıştır. Önerilen algoritma iyi bilinen veri setlerine uygulanarak, sonuçlar değerlendirilmiştir. Sonuçlar literatürce iyi bilinen algoritmalar ile kıyaslanmıştır. Tez kapsamında son yıllarda yayınlanan çalışmalar incelenmiş ve önceki çalışmalara göre daha iyi sonuçlar elde edilmiştir. Algoritmaya ait efektif sonuçlar elde eden sınıflandırıcı seçimi ile büyük boyutlu verilerde daha etkin çalışmasını sağlamak ve algoritmanın kararlılığını arttırmak için kaotik haritalarla birleştirilerek algoritmanın arama yeteneği güçlendirilmiştir. Elde edilen sonuçlar kıyaslanan algoritmalar için tatmin edici sonuçlar üretmiştir. Bu tez çalışmasının en önemli sonucu, optimizasyon tabanlı ikili özellik seçim algoritma modellerinin büyük boyutlu veri için tasarlanarak kaotik haritalarla birleştirildiğinde daha etkili sonuçlar vermesidir.

Önerilen algoritma sürümlerinin umut verici sonuçlarına dayanarak, performansını benzer yöntemlerle karşılaştırmak için yeterince kanıt bulunmaktadır. Nitelik seçme problem süreci çok amaçlı bir problem olarak değerlendirildiğinde sınıflandırma doğruluğunu en üst seviyede, seçilen nitelik sayısını en alt seviyede olması gerekir. Elde edilen sonuçlar tez çalışmasında önerilen yöntemin yüksek rekabetçi sonuçlara

ulaştığını ve çok amaçlı bir optimizasyon problemi olarak ifade edilebileceğini göstermektedir.

#### **1.4. Tez Organizasyonu**

Bu tez çalışması genel hatları ile şu şekilde organize edilmiştir. Birinci bölümde; teze ait genel bakış açısı oluşturmak için tez konusuna ait temel bilgilere yer verilmiş ve tezin amaç, önem ve özgün değeri hakkında detaylı açıklama yapılmıştır. İkinci bölümde; tez çalışması öncesi ve sırasında taranan, literatürde bulunan mevcut çalışmalar ile yapılan kaynak araştırmasına yer verilmiştir. Üçüncü bölümde; tez çalışmasında kullanılan materyal ve metotlara ait kapsamlı bilgi verilmiştir. Yine bu bölümde uygulama aşamasında yapılan yenilik ve gelişmeler, değerlendirme metrikleri ve kullanılan yöntemlerin metodolojik yapısı açıklanmıştır. Dördüncü bölümde; önerilen yöntemle ait yapılan deneysel çalışmalardan elde edilen sonuçlar kapsamlıca açıklanmış, uygulanan istatistiksel yöntemler ile literatür karşılaştırılması yapılarak ulaşılan bulgular yorumlanmıştır. Beşinci ve son bölümde ise tez çalışmasından elde edilen sonuçlara genel bir bakış açısı kazandırılarak gelecek çalışmalar için önerilerde bulunulmuştur.



## 2. KAYNAK ARAŞTIRMASI

Araştırmaya nitelik seçimi problemine çözüm olarak geliştirilen çalışmalar ile başlanmış ardından son yıllarda Metasezgisel algoritmalara artan ilgi nedeniyle nitelik seçimi yapan birçok alternatif algoritma ve yöntemler ile devam edilmiştir.

Karga arama algoritması (CSA), kargalara ait akıllı davranışları modelleyen yeni bir metasezgisel algoritmadır. Uygulama kolaylığı ve sade birleşenleri sebebiyle popüler hale gelmiştir. Bununla birlikte, keşif ve sömürü arasında denge kurabilme kabiliyeti zayıf olduğu için erken yakınsama sorunu bulunmaktadır. Bu nedenle yapılan çalışmada CSA'nın gelişmiş hali özellik seçim yöntemi olarak sarmalayıcı tabanlı kullanılmıştır. ECSA'nın performansı, üç performans metriği ve UCI veri setlerinden 16 veri kümesi kullanılarak istatistiksel önemi ölçülmüştür. Deneysel sonuçlar, önerilen yöntemin daha iyi bir yakınsama hızı ve daha kaliteli bir çözüm sunduğunu göstermiştir (Ouadfel & Abd Elaziz, 2020).

Thuy ve Wongthanavas (2020) yaptıkları çalışmada Kaba Kümeleme (pozitif bölgeye dayalı çalışmalar) ile sezgisel algoritmaların temel hesaplama yöntemlerini birleştirerek güçlü yönlerini kullanmışlardır. Önerilen yöntem büyük ölçekli veri setlerinde hesaplama maliyetini önemli ölçüde azaltmıştır. Büyük veri senaryolarını içeren 26 veri seti üzerinde yapılan çalışmada sonuçlar, önerilen yöntemin hesaplama süresini de etkili bir şekilde azalttığını göstermektedir. Ayrıca, yöntemin performansı Wilcoxon istatistiksel testi kullanılarak doğrulanmıştır (Thuy & Wongthanavas, 2020).

Azar ve ark.'nın (2020) çalışmalarına göre teknoloji geliştikçe, her geçen gün verinin boyutsallığı da artmaktadır. Önerilen çalışmada, Kaba Kümeleme ve Karınca Aslanı Algoritmasına (ALO) dayalı minimum nitelik indigeme için yeni bir arama stratejisini geliştirilmiş ve UCI deposundan veri kümeleri kullanılarak test edilmiştir. İlgili çalışma doğadan ilham alarak karınca aslanlarının avlanma davranışlarını taklit eden ve nitelik seçimi problemine uygulanan bir çalışmadır. Önerilen algoritma, önemli özelliklerin tanımlanmasında iyi bir performans sergilemektedir (Azar vd., 2020).

2019 yılında yapılan başka bir çalışmada Genetik Algoritmanın performansını arttırmak için Destek Vektör Makinesi tabanlı bir yöntem önerilmiştir. Önerilen

algoritmayı kalp hastalığı ile ilişkili veri setlerinde temel nitelik seçim yöntemleri (Relief, Bilgi Kazancı vb.) ile karşılaştırarak, algoritmanın performansının nispeten yüksek olduğu kanıtlanmıştır (Gokulnath & Shantharajah, 2019).

Başka bir çalışmada, nitelik seçimi problemine ait veri seçimi olarak metinler kullanılmış ve Karınca Kolonisi Optimizasyonu (ACO) ve Yapay Sinir Ağı (YSA) tabanlı hibrit bir algoritma önerilmiştir. Önerilen ACO – YSA hibrit algoritması özellik seçim problemi üzerinde verimli bir şekilde çalışmış, büyük veri ortamında uygulanmış ve performansı özellik seçim algoritmaları ile karşılaştırıldığında etkili /optimal bulunmuştur (Manoj vd., 2019).

Yüksek boyutlu veriler üzerinde veri madenciliği yapılmak istendiğinde özellik sayısı arttığından hesaplama maliyeti de aynı düzeyde artar. Kushwaha ve Pant(2018) tarafından yapılan çalışmada, özellik seçim problemini çözmek için K-kümeleme stratejisinin yeni bir ikili Parçacık Sürüsü Optimizasyonu ile entegrasyonu önerilmiştir. LBPSO algoritması, orijinal metin veri kümesini girdi olarak alır ve önemli özelliklerin yeni alt kümesini üretir. K-kümeleme algoritması, özellik seçme yöntemini değerlendirmek için bu özellikleri girdi olarak alır. Deneysel sonuçlar ayrıca LBPSO'nun metin kümeleme için diğer BPSO tabanlı nitelik seçimi yöntemlerinden daha etkili olduğunu doğrulamaktadır (Kushwaha & Pant, 2018).

Anaraki ve ark(2018). yaptığı çalışmada, özellik azaltmayı etkin bir şekilde gerçekleştiren ve böylece bir veri kümesinin en bilgilendirici özelliklerini seçen yeni bir özellik seçme yöntemi sunulmuştur. İlgili yöntem iki bileşenden oluşur: 1) özellik alt kümesi değerlendirmesi için bir ölçü belirler ve 2) bir arama stratejisi ile algoritmayı tamamlar. Değerlendirme ölçüsü olarak bulanık kaba kümeleme önerilirken, arama stratejisine olarak ikili Kurbağa Sıçraması Algoritmasının yeni bir versiyonu önerilmektedir (B-SFLA). Parametrik olmayan istatistiksel testler, önerilen yaklaşımı UCI veri setlerinden dokuz yüksek boyutlu ve büyük olanlar da dâhil olmak üzere toplam yirmi iki veri seti üzerinde mevcut birkaç yöntemle karşılaştırılmıştır. Deneysel sonuçlar, B-SFLA yaklaşımının seçilen özelliklerin sayısı ve sınıflandırma doğruluğu açısından diğer meta-sezgisel yöntemlerden önemli ölçüde daha iyi performans gösterdiğini göstermektedir (Anaraki vd., 2018).

Chen ve Kanoh'a ait başka bir çalışmada, Hamming mesafe ölçümü tabanlı bir yöntem önerilmiştir. Hesaplanan benzerlik oranına göre kaynak vektörde bit değişimi

yapılarak aday çözüm üretilmiştir. Önerilen yöntem graf boyama probleminde kullanılmıştır (K. Chen & Kanoh, 2016).

Yapay Arı Kolonisi (ABC) algoritması çeşitli problemlere uygulanan popüler bir algoritmadır. Jia ve ark.(2014) vektörleri ikili uzayda tanımlayarak, vektörün her bir boyutu için  $[0,1]$  aralığında rastgele bir sayı üretmiş ve bu sayı 0,5'den küçükse, ilgili parametreye 0, büyükse 1 değerini atamıştır. Mevcut kaynaklardan yeni çözümler üretebilmek için 'xor', 'and' ve 'or' bit işlem operatörlerini kullanmışlardır. ABC algoritmasının ikili bir versiyonu (BitABC) olarak önerilen yöntem en çok kullanılan ikili arama tekniği Genetik Algoritma ile karşılaştırılarak 13 fonksiyonun optimizasyonu için uygulanmıştır (Jia vd., 2014).

Yine ABC algoritması ile yapılan başka bir çalışmada, algoritmanın ikili bir versiyonu önerilmiştir. Orijinal ABC algoritmasından farklı olarak, önerilen algoritma, görevli arıların, gözcülerin ve izcilerin yiyecek arama yörüngelerini değiştirmek için ayrı operatörler uygulamıştır. Dört matematiksel kıyaslama işleviyle önerilen algoritma genetik algoritma (GA) ve parçacık sürüsü optimizasyonundan (PSO) önemli ölçüde daha iyi performansa sahip olduğu kanıtlanmıştır (K. Chen & Kanoh, 2016).

Tubishat ve ark.(2020) tarafından özellik seçim problemlerini çözmek için sarmalayıcı tabanlı ve Salp Sürü Algoritmasının geliştirilmiş bir versiyonu olan çalışmada (ISSA), SSA algoritmasının zayıf yönlerinin iyileştirilmesi önerilmiştir. Önerilen algorithma mevcut arama yöntemlerine ek olarak arama alanında iki yönlü tarama stratejisi benimsenmiş ve en uygun çözüm ile bir sonraki iterasyona geçiş sağlanmıştır. Çalışmanın, Genetik Algoritma, Parçacık Sürüsü Optimizasyonu, Grasshopper Optimizasyon Algoritması ve Karınca Aslanı Optimizasyonu gibi dört iyi bilinen optimizasyon algoritmasıyla karşılaştırılarak daha iyi performans gösterdiği ortaya çıkmıştır(Tubishat vd., 2020). Ardından 2021 yılında yapılan çalışmayı geliştirmek ve SSA'nın sorunlarını çözmek için kaotik haritalama yöntemi üzerine iki ana iyileştirme dahil edilerek algoritmanın performansı ölçülmüştür. İlk iyileştirme, salplerin pozisyon güncellemesi ve SSA çözümlerinin çeşitliliğini artırmak için Singer'in Kaotik Haritası kullanılarak yeni bir denklemin geliştirilmesini içerir. Bir diğer iyileştirme yöntemi olarak SSA yeni bir yerel arama algoritması (LSA) ile birleştirilir. Önerilen DSSA, bir sarmalayıcı modunda k-NN sınıflandırıcısı ile birleştirilerek, UCI havuzundan 20 karşılaştırma veri kümesi ve 3 Hadith veri kümesi için çalıştırılmıştır. DSSA sonuçları, orijinal SSA ve Parçacık Sürüsü Optimizasyonu (PSO), Genetik

Algoritma (GA), Karınca Aslanı Optimizasyonu (ALO) ve Grasshopper Optimizasyon Algoritması (GOA) dahil olmak üzere dört iyi bilinen optimizasyon algoritmasıyla karşılaştırılmıştır. DSSA doğruluk, uygunluk fonksiyonu, seçilen nitelik sayısı açısından 23 veri seti ile kıyaslanmıştır. Önerilen algoritmanın Orijinal SSA ve iyi bilinen optimizasyon algoritmalarından performansının daha yüksek olduğu ortaya çıkmıştır (Tubishat vd., 2021).

Jaya Optimizasyon Algoritması kullanılarak tasarlanan başka bir çalışmada, çok amaçlı yeni bir özellik seçme algoritması önerilmektedir. Sınıflandırma doğruluğu ve nitelik seçiminin en az olması, birlikte optimize edilmesi gereken çok amaçlı hedeflerdir. Önerilen çalışma optimizasyon algoritmalarının erken yakınsama problemine farklı bir çözüm getirmiş, akciğer kanseri, meme kanseri, diyabet, doğurganlık ve immünoterapi gibi beş veri seti için uygulanmış ve sonuçlar, diğer mevcut tekniklerle karşılaştırılmıştır. Önerilen algoritma, veri madenciliği tabanlı tıbbi teşhis karar destek sisteminin etkinliğini artırmaktadır (Sheth vd., 2020).

2020 yılında yapılan başka bir çalışmada Yapay Kelebek Optimizasyonunun (ABO) tekli, çoklu ve çok amaçlı versiyonları nitelik seçim problemine uygulanmıştır. Ayrıca geliştirilen algoritma için her bir sınıfın sınıflandırma doğruluğunu ayrı ayrı optimize eden ve özellik setinin minimizasyonunu sağlayan iki farklı yaklaşım önerilmiştir. Önerilen yaklaşımlar, UCI'den alınan sekiz veri kümesi üzerinde iyi bilinen Parçacık Sürüsü Optimizasyonu, Ateşböceği Algoritması, Çiçek Tozlaşma Algoritması, Beyin Fırtınası Optimizasyonu ve Kara Delik Algoritması ile karşılaştırılmış, ikili ABO'nun diğer meta-sezgisel tekniklerden daha iyi performans gösterdiği, daha az özellik seçtiğini ve daha düşük bir hesaplama yükü oluşturduğunu göstermiştir (Rodrigues vd., 2020).

2021 yılında yapılan bir çalışmada tahmin işlemi için iki katmanlı bir özellik seçme yaklaşımı önerilmiştir. Önerilen yöntemin ilk katmanında, Genetik Algoritma (GA), nitelik sayısını ve tahmin hatasını azaltmayı amaçlayan, verilerin optimal alt kümesini aramak için bir sarmalayıcı metod tabanlı tasarlanırken, GA'nın hesaplama verimliliği nedeniyle seçilmesine rağmen optimallliği garanti edemediği tespit edilmiştir. Bu sorunu çözebilmek, tahmin doğruluğunu iyileştirmek ve nitelik seçimi yapabilmek için önerilen yöntem Esnek Ağ/Elastic Net(EN) kullanılarak ikinci bir katman eklenir. Esnek Ağ esnekliği ve zaman verimliliği nedeniyle ikinci katmanda gömülü yöntem olarak seçilmiştir. Bu iki katmanlı yaklaşım, tahmin sayısının gözlem sayısına göre

farklı oranlara sahip olduğu çok sayıda veri alt kümesinden oluşan Maize genetik veri kümesine uygulanmıştır. Elde edilen sonuçlar önerilen modelin daha iyi sonuçlar ürettiğini doğrulamaktadır. Bu çalışmanın diğerlerinden farkı önce sarmalayıcı yöntem kullanarak nitelik seçimi yapmış olup ardından filtre tabanlı bir gömülü bir yöntemle süreci tamamlamasıdır(Amini & Hu, 2021).

Başka bir çalışmada Yapay Sinir Ağında (YSA) etkili özellik seçimi ve parametre optimizasyonunun paralel olarak işlenmesi için Entegre Yapay Bağışıklık sistemi ve Yapay Arı Kolonisine dayalı meme kanseri teşhisi önerilmiştir. Yapay sinir ağının momentuma dayalı gradyan inişi kullanılarak geliştirilen algoritmanın yapay sinir ağının esnek geri yayılma tekniği ile % 23'lük daha iyi hassasiyet veya gerçek pozitif oran verdiği ortaya çıkmıştır. Geliştirilen strateji özellik seçimi ve yapay sinir ağının başlangıç ağırlıklarının tahmin edilmesi üzerine kurulmuştur. Yerel arama sürecini iyileştirmek için Tavlama Benzetimi Algoritmasının (SA) avantajlarını kullanan algoritmanın Wilcoxon yöntemiyle YSA'da % 99,34 ve % 99,11'lik bir ortalama doğrulama ile sınıflandırma ürettiği doğrulanmıştır. Önerilen algoritmanın, meme kanserinin kesin teşhisini yüksek oranda yapabildiği kanıtlanmıştır (S. vd., 2021).

İkili kimyasal reaksiyon optimizasyonu (BCRO) ve ikili parçacık sürü optimizasyonu (BPSO) tabanlı geliştirilen Kimyasal reaksiyon (BCRO) ve parçacık sürü optimizasyonlarının ikili versiyonu ile oluşturulan bir başka algoritma (HBCRO-BPSO), sınıflandırma doğruluğu ve özellik seçimini optimize etmek amaçlı geliştirilmiştir. Geliştirilen algoritmaların BPSO, gelişmiş ikili karınca kolonisi optimizasyonu (ABACO), ikili genetik algoritma (BGA) ile performansları karşılaştırılmıştır. K-en yakın komşu (k-NN), Lojistik Regresyon, Naive Bayes, Karar Ağacı ve Rastgele Orman dahil olmak üzere beş Makine Öğrenmesi algoritması ile sınıflandırma doğruluğu ölçülmüştür. UCI deposundan on bir veri kümesi üzerinde test edilen deneysel sonuçlar, önerilen algoritmanın özellik azalmasının ve doğruluk iyileştirmesinin sırasıyla % 5,01 ve % 3,83 artırdığını göstermektedir. Geliştirilen HBCRO- BPSO tabanlı özellik seçme yöntemi; BGA'ya göre %4.58 ve %3.12; ve bir k-NN sınıflandırıcısı ile kullanıldığında ise ABACO'ya göre %4.15 ve %2.27 daha iyi performans göstermiştir (Srinivasa Rao vd., 2020).

Sosyal örümcek algoritması (SSA), sürekli problemleri çözmek amacıyla örümcek davranışları üzerinde oluşturulan sezgisel bir algoritmadır. Baş ve Ülker(2020) tarafından yapılan çalışmada ikili Sosyal Örümcek Algoritması (BinSSA)

önerilmiştir. Çalışmada sekiz adet S-şekilli ve V-şekilli transfer fonksiyonları kullanılmıştır. SSA'nın sürekli uzaydan ayrık uzaya dönüşümü için transfer fonksiyonları kullanılmış ve BinSSA elde edilmiştir. Böylece BinSSA1, BinSSA2, BinSSA3, BinSSA4, BinSSA5, BinSSA6, BinSSA7 ve BinSSA8 olarak sekiz farklı BinSSA varyasyonu oluşturulur. BinSSA'nın araştırılması ve kullanılması için, ilgili algoritmaya BinSSA-CR olarak bir çapraz geçiş operatörü eklenmiştir. BinSSA varyasyonları özellik seçimi problemine uygulanmıştır. K-en yakın komşu (k-NN) ve destek vektör makineleri (SVM) sınıflandırma algoritması olarak kullanılmıştır. BinSSA, düşük, orta ve büyük ölçekli yirmi bir UCI veri setinde değerlendirilmekte ve elde edilen sonuçlar literatürdeki iyi bilinen algoritmalarla karşılaştırılmaktadır. Deneysel sonuçlar, BinSSA ve BinSSA-CR'nin kıyaslanan algoritmalara göre daha iyi performans gösterdiğini ispatlamaktadır. Çalışmada özellik seçme problemi için hem çapraz operatör kullanılarak (BinSSA-CR) hem de çapraz operatör (BinSSA) olmadan geliştirilen ikili sosyal örümcek algoritması önerilmiştir (Baş & Ülker, 2020).

Veri madenciliğinde özellik seçiminin önemini göstermek ve en iyi özellik seçme yöntemini sınıflandırma tekniğini ile bulmak amaçlı yapılan başka bir çalışmada birkaç sınıflandırma yöntemi arasından Karar Ağacı, Bayes ağları, K-En Yakın Komşu, Vaka Tabanlı Muhakeme, Genetik Algoritma ve Bulanık Mantık teknikleri kıyaslanmıştır. Çalışma özellik seçim yöntemlerini ve sınıflandırma tekniklerini tüm yönleriyle incelemiştir. Karşılaştırılan öğrenme algoritmalarından C4.5 algoritmasının (Karar Ağacı), Naive Bayes ve K-en yakın komşu algoritmalarına kıyasla özellik sınıflandırma teknikleri için en iyi sonuçları verdiği tespit edilmiştir (Rajabhushanam, 2020).

Genetik Algoritma ve PSO'ya dayalı iki yeni özellik seçim yaklaşımı Hichem ve ark.(2018) geliştirilmiştir. Çalışmada on iki kıyaslama veri seti özellik seçim probleminin temsili örnekleri olarak farklı sayıda özelliğe (10'dan 617'ye kadar), sınıflara ve örneklere sahip olan UCI makine öğrenimi deposundan seçilmiştir. Her veri kümesi için tüm örnekler eğitim ve test seti olarak ayrılmış, sınıflandırma algoritması olarak ise K-en yakın komşu (K-NN) öğrenme algoritması kullanılmıştır. Çalışma sonucunda daha az sayıda özellik seçimi ile daha iyi sınıflandırma doğruluğu elde edilmiştir (Hichem vd., 2018).

Teknolojinin gelişmesiyle tüm verilerde olduğu gibi tıbbi konulardaki üretilen verilerde büyük boyutlu hale gelmiştir. Özellik seçme yöntemi, büyük boyutlu

verilerdeki karmaşıklığın çözümü için kullanılabilir. Agustian ve Lubis(2020) tarafından yapılan çalışmada Particle Swarm Optimization (PSO) algoritması özellik seçimi için kullanılmıştır. SVM, Naive Bayes, Logistic Regression, Decision Tree ve K-NN gibi sınıflandırma algoritmalarının performansı uygunluk fonksiyonu olarak değerlendirilmiş, ayrıca PSO performansını Genetik Algoritma ile karşılaştırmışlardır. Yapılan testlerde Parçacık Sürüsü Optimizasyonunun çeşitli sınıflandırma algoritması performanslarını iyileştirebileceğini kanıtlamaktadır. Ancak PSO'nun, özellik seçimi olarak Genetik Algoritmanın performansından daha etkili olmadığı ortaya çıkmıştır. PSO ile birleştirilen Naive Bayes kıyaslanan algoritmalara göre daha etkili sonuçlar üretmiştir. Ancak SVM PSO için en yüksek doğruluk sonuçlarına sahip sınıflandırma algoritması olarak tespit edilmiştir (Agustian & Lubis, 2020).

Penguenlerin rastgele gezerek yiyecek arama davranışından esinlenilerek geliştirilen Penguen Arama Optimizasyon algoritması kullanılarak özellik alt kümesi indirgemesi için yapılan başka bir çalışmada, algoritmanın gerçekleştirdiği her durum için üç farklı sınıflandırıcı uygunluk değerini oluşturulmuştur. Bu sınıflandırıcılar Random Forest, Nearest Neighbor ve SVM 'dir. Bununla birlikte, önerilen yaklaşım UCI havuzundan toplanan iyi bilinen bazı kıyaslama veri kümelerinde uygulanmış, sınıflandırma doğruluğunu son teknoloji algoritmalarla karşılaştırılmıştır. Bu çalışmanın diğer çalışmalardan farkı önerilen özellik seçimi algoritmasını PCA ve LDA ile de karşılaştırılmasıdır. Çalışmada önerilen algoritmanın kıyas algoritmalarından daha iyi sonuçlar verdiği ortaya çıkmıştır (Dasgupta vd., 2019).

İkili çok amaçlı ABC algoritmasının önerildiği başka bir çalışmada (Bin-MOABC ve Num-MOABC) algoritmaların performans analizi, tek hedefli ABC algoritmaları (ABC-ER ve ABC-Fit2C), geleneksel algoritmalar (LFS ve GSBS) ve çok amaçlı algoritmalar (NSGAIL, NSSABC ve MOPSO) olmak üzere 12 kıyaslama veri seti üzerinden karşılaştırmalar yapılmıştır. Elde edilen sonuçlar boyut indirgeme ve sınıflandırma doğruluğu açısından Bin-MOABC'nin kıyaslanan yöntemlerden daha etkili sonuçlar ürettiğini göstermektedir. Önerilen algoritmalar çok amaçlı olduğundan özellik seçim problemi için uygulanabilir. Önerilen algoritmanın hesaplama maliyeti yüksek olmasına rağmen, yüksek boyutlu veri kümelerinde ölçeklenebilirliği denenebileceği tavsiye edilmiştir (Hancer vd., 2018).

Son yıllarda, web kimlik avı saldırısı, kimlik avcılarının finansal hırsızlık yapmak için internet kullanıcıları hakkında önemli finansal bilgileri çalabildiği en ciddi web

güvenliği sorunlarından biri haline gelmiştir. Ali ve Ahmed (2019) tarafından yapılan çalışmada, kimlik avı web sitesi tahminini geliştirmek için, evrimsel algoritmaya dayalı özellik seçimi ve ağırlıklandırma yöntemleriyle derin sinir ağlarını (DNN) kullanan hibrit akıllı kimlik avı web sitesi tahmini önerilmektedir. Önerilen yaklaşımda, özelliklerin optimum ağırlıkları, kimlik avı web sitesi tahmininin doğruluğunu artırmaya yardımcı olmak için genetik algoritma (GA) ile sezgisel olarak tanımlanır. GA'nın seçtiği web sitesi özellikleri DNN'i eğitmek ve tahmin yapmak üzere kullanılmıştır. Elde edilen sonuçlar önerilen çalışmanın kıyaslanan kimlik avı web sitesi tahminlerinden daha iyi sonuçlar ürettiğini göstermiştir (Ali & Ahmed, 2019).

COVID-19 akciğer semptomlu ve ölümcül bir hastalıktır. Yapılan çalışmada COVID-19 semptomları gösteren hastalardan alınan akciğer X-ışını görüntülerinden oluşturulan veri seti kullanılmıştır. Görüntüler gerekli ön işlemlerden geçtikten sonra AlexNet, VGG19, GoogleNet ve ResNet gibi derin öğrenme modelleri kullanılarak veri setlerinden özellik çıkarma işlemi uygulanmıştır. Ardından en iyi özellik seçimini elde edebilmek için parçacık sürüsü ve gri kurt optimizasyon algoritmalarının ikili versiyonları kullanılmıştır. Geliştirilen veri setinin SVM kullanılarak sınıflandırılması sonucu genel doğruluğu % 99,38 olarak elde edilmiştir. Deneysel sonuçlar, önerilen yaklaşımın COVID-19 teşhis çalışmaları sırasında uzmanlara yardımcı olabileceğini kanıtlanmıştır (Canayaz, 2021).

Özellik seçimi (FS), bir özellik vektöründen tüm olası alakasız ve gereksiz özellikleri kaldırmayı ve böylece genel tahmin veya sınıflandırma modelinin performansını artırmayı amaçlar. Ghosh ve ark.(2020) tarafından yapılan çalışmada, FS problemlerini çözmek için İkili Yelken Balığı (BSF) olarak adlandırılan ve yakın zamanda önerilen Yelken Balığı Optimizasyonun (SFO) ikili bir varyantı geliştirilmiştir. SFO Sigmoid transfer fonksiyonu kullanılarak ayrıklaştırılmıştır. İkili SFO'nun kullanım yeteneğini geliştirmek için Adaptif  $\beta$ -tepe tırmanışı (A $\beta$ HC) ile BSF optimize ediciyi birleştirilerek, İkili SFO ve ikili SFO algoritmaları 18 farklı veri kümesine uygulanmış, 10 adet iyi bilinen meta-sezgisel nitelik seçimi yöntemiyle karşılaştırılmıştır. Deneysel sonuçlar, önerilen yöntemlerin kıyaslanan algoritmalara göre daha üstün olduğunu göstermektedir (Ghosh vd., 2020).

Saldırı tespit sistemlerinin analizinde derin öğrenme algoritmaları ve makine öğrenmesi kullanılmaktadır. Bu amaçla yapılan başka bir çalışma da ise, Kaggle



deposundan toplanan bir NSL-KDD ve KDD Cup 99 veri kümesi, min-maks normalizasyon tekniği ve 1-N kodlama yönteminde işlenerek, nitelik indirgeme problemi için bir Örümcek Maymunu Optimizasyonu (SMO) algoritması ile kullanmış ve indirgenmiş veri seti derin bir sinir ağı (DNN) girişine uygulanmıştır. Model ayrıca temel bileşen analizine (PCA) dayalı DNN ve klasik DNN modelleriyle karşılaştırılarak elde edilen sonuçların önerilen modeli uygulayarak diğer yaklaşımlara göre avantajını doğrulamıştır (Khare vd., 2020).

Tıbbi verilerden tümör sınıflandırılmaları kullanılarak elde edilen tanı geliştirme süreçlerinde DNA mikro dizi verileri büyük önem taşımaktadır. Yüksek boyutlu veriler ile tümör sınıflandırması amacıyla önemli genleri seçmek için günümüzde birçok araştırmacı, çeşitli gen seçme yöntemlerini kullanarak mikro dizi verilerini araştırmaktadır. Shukla ve ark. tarafından önerilen yöntem, ilk aşamada iki aşamalı, çok katmanlı yaklaşım kullanan EGS metodundan oluşur ve veri setinden gürültülü/gereksiz genleri filtrelemek için f-skor yaklaşımı uygulanır. Uyarlanabilir Genetik Algoritma (AGA), sarmalayıcı model Destek vektör makinesi (SVM) ve Naive Bayes (NB) sınıflandırıcılarını gen alt kümelerini tanımlamak için kullanılmıştır. Ayırt edici genleri yüksek sınıflandırma doğruluğu ile seçmeyi amaçlayan algoritmaya ait deneysel sonuçlar, önerilen yöntemin, kardinalitede önemli bir azalmaya destek sağladığını, doğruluk ve optimal sayıda gen açısından son teknoloji gen seçme yöntemlerinden daha iyi performans gösterdiğini göstermektedir (Shukla vd., 2018).

Başka bir çalışmada literatürde sıkça rastlanılan Yusufçuk Algoritmasının arama parametrelerine kaotik haritaların uygulandığı yeni bir Kaotik Yusufçuk Algoritması (CDA) önerilmiştir. Yusufçuk Algoritmasının (DA) yakınsama oranını hızlandırmak, verimliliğini artırmak ve optimizasyon süreci boyunca yusufçukların hareketlerinin ana parametrelerini ayarlamak için on kaotik harita kullanılmıştır. Önerilen modelde CDA kullanılarak özellik seçimi yapılmıştır. Ardından CDA tarafından belirlenen özellikler Destek Vektör Makinesi (SVM) sınıflandırıcısı ile doğrulanmıştır. Önerilen algoritma, 6712 ilaca ait veri setlerini seçmek için kullanılmıştır. Deneysel sonuçlar, Gauss kaotik haritasının DA'nın performansını önemli ölçüde arttırdığını, CDA'nın sınıflandırma performansını en üst düzeye çıkararak ve seçilen özelliklerin sayısını DA ve diğer meta-sezgisel optimizasyon algoritmalarına kıyasla en aza indiren optimum özellik alt kümesini bulma yeteneğini kanıtlamıştır (Sayed, Tharwat, vd., 2019).

Sınıflandırma doğruluğu, öğrenilmiş bir model oluşturmak için kullanılan girdi özelliklerinin kalitesinden büyük ölçüde etkilenir. Özellik seçimi, orijinal veri kümesinden en etkili özellik alt kümesini seçmeyi amaçlayan önemli bir ön işleme görevidir. Bu nedenle, özellik seçim yönteminin kullanılması, sınıflandırma doğruluğunu artırmak ve inşa edilen modelin karmaşıklığını azaltmak için gereklidir. Yapılan çalışmada, WBSCA adlı özellik seçimi için sarmalayıcı tabanlı ikili Sinüs Kosinüs Algoritması (SCA) önerilmektedir. Önerilen algoritma, UCI makine öğrenimi havuzundaki yedi sınıflandırma veri kümesi üzerinden iyi bilinen üç ikili algoritma ile (BBA, BGSA, BDA) karşılaştırılmıştır. Sonuçlar, önerilen algoritmanın optimum alt kümeyi ararken ve özelliği seçerken daha iyi performans gösterdiğini ortaya koymuştur (Taghian & Nadimi-Shahraki, 2019).

Yakın zamanda ortaya çıkarılan Harris Hawk Optimizasyon (HHO) Algoritmasının ikili bir versiyonun kullanıldığı Piri ve Mohapatra (2021) tarafından yapılan başka bir çalışmada, optimal öznitelik alt kümelerini oluşturmak için sarmalayıcı sınıflandırıcı olarak K-En Yakın Komşu (k-NN) yöntemi ve çok amaçlı Kuadratik İkili HHO (MOQBHHO) algoritmasına ait yaklaşımın performansını tahmin etmek için on iki standart tıbbi veri seti kullanılmıştır. Önerilen MOQBHHO algoritması, MOBHHO-S (bir sigmoid fonksiyonu kullanarak), çok amaçlı genetik algoritma (MOGA), Çok Amaçlı Karınca Aslan Optimizasyonu (MOALO) ve NSGA-II ile karşılaştırılmıştır. Karşılaştırmalar, önerilen algoritmanın özellik alt kümelerini etkili şekilde indirgediğini ortaya çıkarmıştır (Piri & Mohapatra, 2021).

2021 yılında sürü tabanlı stokastik algoritma olan Sümüksü Kalıp Algoritması (SMA)'dan geliştirilerek önerilen (DFSMA) algoritması tanıtılmıştır. IEEE CEC2017'de kullanılan algoritma, 11 metasezgisel algoritma, 10 geliştirilmiş algoritma ve yakın zamanda tanıtılmış 3 farklı algoritma ile karşılaştırılmıştır. Karşılaştırmalar neticesinde DFSMA daha iyi sonuçlar üretmiştir. Ek olarak, ikili DFSMA (BDFSMA) elde edilmiş, BDFSMA'nın performansı UCI deposundaki 12 veri seti üzerinde değerlendirilmiştir. Deneysel sonuçlar, BDFSMA'nın orijinal SMA'dan daha iyi performans gösterdiğini ve diğer optimizasyon algoritmaları ile karşılaştırıldığında, sınıflandırma doğruluğunu iyileştirip seçilen özelliklerin sayısını azalttığını ortaya çıkarmıştır (J. Hu vd., 2021).

Filtre ve sarmalayıcı uygunluk fonksiyonlarının aynı anda optimize edildiği çok amaçlı Balina Optimizasyon Algoritması (WOA) kullanılarak önerilen başka bir

algoritma, iyi bilinen yedi algoritma ve on iki kıyaslama veri kümesi üzerinde test edilmiştir. Deneysel sonuçlar, önerilen algoritmanın, mükemmel sınıflandırma doğruluğu ile daha az sayıda öznitelik içeren birkaç alt küme elde etme yeteneğini göstermektedir (Got vd., 2021).

Sarmalayıcı tabanlı oluşturulan başka bir çalışmada erken yakınsama nedeniyle elde edilen alt-optimal çözümleri sıfırlamak için açgözlü çaprazlamalı çok amaçlı bir İkili Yarasa Algoritması (BBA) önerilmiştir. Seçilen özniteliklerin değerlendirilmesi, 10 kat çapraz doğrulama ile Destek Vektör Makinesi kullanılarak yapılmış, UCI deposunda bulunan kıyaslama veri kümeleri ile değerlendirilmiştir. Avustralya, Ionosphere, Wisconsin Meme Kanseri ve Musk kullanılarak %89.25, %96.45, %96.57 ve %88.50 sınıflandırma doğruluğu elde edilmiştir. Açgözlü çaprazlama ile önerilen çok amaçlı İkili Yarasa Algoritması, mevcut yarasa tabanlı algoritmalar ve diğer doğadan ilham alan algoritmalar üzerinde daha iyi performans sağladığı ortaya çıkmıştır (Akila & Allin Christie, 2022).

GR-MOEA adlı etkileşimli bir filtre-sarmalayıcı çok amaçlı evrimsel algoritma 2021 yılında önerilmiştir. Algoritma da diğer algoritmalarından farklı olarak iki popülasyon (filtre popülasyonu ve sarmalayıcı popülasyon) eşzamanlı olarak gelişir ve evrim sırasında daha yüksek kalitede nihai özellik alt kümesini elde etmek için birbirleriyle etkileşime girerler. Farklı veri kümeleri üzerindeki karşılaştırma sonuçları, GR-MOEA'nın hem doğruluk hem de seçilen özelliklerin sayısı açısından mevcut en son teknolojiye göre üstünlüğünü göstermiştir (Z. Liu vd., 2021).

Sarmalayıcı metasezgisel yöntemlerde yüksek boyutlu veri kümeleri etkinliğinin azaldığı ve hesaplama maliyetleri arttığı literatürce de bilinen bir gerçektir. Yapılan çalışmada bu sorunu çözmek için, ReliefF filtre yöntemi ve yeni bir meta-sezgisel Denge Optimize Edici'ye (EO) dayalı hibrit bir nitelik seçimi yaklaşımı önerilmiştir. Önerilen RBEO-LS yöntemi iki aşamalıdır. İlk aşamada, ReliefF algoritması, sınıflandırma göreviyle verilerin birbirleri ile ilişkisini tahmin ederek özelliklere ağırlıklar atayarak bir ön işleme adımı gibi kullanılır. Ardından İkili EO (BEO) sarmalayıcı yaklaşımı kullanarak özellikleri ağırlıklarına göre sıralar ve BEO popülasyonunu başlatır. Algoritmaya ait performans UCI veri setinden alınan yüksek boyutlu veri setleri üzerinde değerlendirilmiştir. Sonuçlar, ReliefF algoritması ile İkili EO algoritması kullanımının, EO algoritmasının performansını iyileştirdiğini ve diğer

son teknoloji yaklaşımlar arasında RBEO-LS'nin üstünlüğünü de göstermektedir (Ouadfel & Abd Elaziz, 2022).

Orman Optimizasyon algoritmasına (FOA) dayalı tasarlanan başka bir çalışmada özellik seçim algoritması için sürekli ve ikili gösterimler kullanılarak iki versiyon geliştirilmiştir. Algoritmaya ait performans ilk aşamada dokuz UCI veri seti ile ardından çeşitli tek ve çok amaçlı yöntemle karşılaştırılmıştır. Sonuçlara göre önerilen her iki algoritma da diğer çok amaçlı amaçlarla karşılaştırıldığında aynı performansa ulaşmış hatta tek amaçlı yöntemlerden daha iyi performans göstermiştir. Sürekli MOFOA, diğer yöntemlere göre daha az sayıda öznelik seçerek çoğu durumda sınıflandırma hatasını azaltmayı başarmıştır (Nouri-Moghaddam vd., 2021).

İkili Grup Arama Optimize Edici'yi (GSO) sarmalayıcı tabanlı özellik seçim yöntemi olarak öneren Abualigah ve Diabat (2022) algoritmayı Kaotik haritalar ile geliştirmişlerdir. Önerilen yöntemde GSO algoritması Lojistik, Parçalı, Singer, Sinüzoidal ve Tent Kaotik Haritaları ile birleştirilmiştir. GSO algoritması k-NN sınıflandırma algoritmasını bir tümevarım algoritması gibi kullanır. Üç amaçlı önerilen algoritmaya ait amaç fonksiyonu en iyi sınıflandırma doğruluğu, en az seçilen özellik sayısı ve model karmaşıklığının en aza indirgenmesini sağlamak üzere kurgulanmıştır. Deneysel sonuçlar yirmi iyi bilinen UCI veri seti üzerinde literatürce iyi bilinen yöntemler ile kıyaslanmış ve özellikle Tent Kaotik Haritası ile kullanılan ikili GSO'nun diğer yöntemlerden daha iyi performans gösterdiği ortaya çıkmıştır (Abualigah & Diabat, 2022).

Yapılan başka bir çalışmada sınıflandırma doğruluğunu iyileştirmek için önce ikili Karahindiba Algoritması (BDA) önerilmiş, ardından algoritmanın performansını iyileştirmek için, geliştirilmiş bir tohumlama stratejisi kullanan ikili karahindiba algoritması kaotik popülasyonları (SBDA) önerilmiştir. İlk aşamada tohumlama alanını optimal olarak kullanma stratejisi algoritmanın arama uzayı performansını artırır. İkinci olarak, tohumlar üretilirken kaotik operatörler kullanılarak kaotik popülasyonlar oluşturulur, bu da algoritmanın yerel optimumdan dışarı atlama yeteneğini geliştirir ve algoritmanın kararlılığını geliştirir. Çalışmada veri tabanlarından toplanan 15 veri seti kullanılarak algoritmanın performansı doğrulandı. Elde edilen sonuçlar önerilen SBDA'nın performansının yüksek sınıflandırma doğruluğu ile daha az nitelik seçimi yaptığı ve kıyaslanan algoritmalarından daha iyi olduğunu göstermektedir (Zhao vd., 2022).

Stokastik yöntemlere dayanarak öznitelik seçiminde birçok algoritma kullanılmıştır. Bunlardan bazıları ise; Karınca Kolonisi Optimizasyonu (ACO) (Guntch & Middendorf, 2002), Balık Sürü Algoritması (FSA)(Li, 2002), Diferansiyel Evrim Algoritması (DE), Beyin Fırtınası Algoritması (BSO)(Shi, 2015) , Ay Alevi Optimizasyonu (MFO)(Mirjalili, 2015) vb.'dir.

İlgili çalışmaların bir sonucu olarak, metasezgisel algoritmaların özellik azaltmada arama stratejisi olarak kullanılabilir iyi teknikler olduğunu dikkate almak için yeterli kanıt vardır. Ayrıca literatürdeki son çalışmalar incelendiğinde özellikle metasezgisel yaklaşımların performansları nedeniyle çeşitli problemler için kullanılmasının artarak popüler hale geldiği görülmektedir. Belirli bir veri kümesinin sadece bir kısmını işleyerek iyi sonuçlara ulaşılması ve tüm veri kümesinin bir bilgisayarın belleğine sığacak kadar küçük olmasını gerektirmemesi sonucunun, büyük olan veri kümelerini analiz edenler için önemli olduğunu göstermektedir.



### 3. MATERYAL VE YÖNTEM

Benchmarking ifadesi, “kıyaslama” veya “referans noktasının belirlenmesi” anlamına gelmektedir. Önerilen yöntemin rekabet gücünü ortaya koymak amacıyla, aynı problemin çözümüne yönelik geliştirilen mevcut başarılı yöntemlerle incelenmiş, performanslardaki farklılığı yaratan özelliklerin neler olduğu araştırılmıştır. Bu kıyaslama, geliştirilen yöntemin rakipleri arasındaki mevcut durumunu göstererek geliştiricinin yöntemin zayıf ve güçlü yanlarını tespit edebilmesini sağlamaktadır. Geliştirilen yöntemin tercih edilmesi için diğerlerinden üstün veya farklı olması gerekir. Bu farkı göstermek için benchmark testi yapılır. Problemden geliştirilen yöntemin kıyaslanabilmesi için uygulama alanının temel özelliklerini içeren farklı boyut ve nitelikte veriler kullanılmıştır.

Tez kapsamında amaç yeni çözüm yaklaşımları geliştirmek olduğundan, gerçek dünya problemleri yerine yöntemlerin performans analizi amacıyla literatürde öznitelik indirgeme problemi için kullanılan benchmark veri setleri ve iyi bilinen algoritmalar kullanılmıştır. Böylelikle kıyaslama yaparken geliştirilecek yöntemlerin her kategorideki performansını değerlendirmek yöntemin etkinliğini ve kalitesini göstermek açısından faydalı olacaktır.

#### 3.1. Kaos Teoremi

Kaos teoremi ilk olarak 1963 yılında matematikçi Edward Lorenz tarafından ortaya atılan ve kaosuun özelliklerini açıklamak için kullanılan bir kuramdır. Lorenz, hava durumu tahmini yapmak için kullanılan bir modeli incelediğinde, sistemin çok küçük değişikliklerle bile çok büyük farklılıklar gösterebileceğini keşfetmiştir. Bu sonucu, "butterfly effect" olarak adlandırmış ve bu etki kaosu teoreminin temel prensiplerinden biri olarak kabul edilmiştir.

Teoremin temel fikri, bir sistemin küçük değişikliklerinin büyük sonuçlar doğurabileceğini içermektedir. Bu durum, sistemin öngörülemezliğini ve tahmin edilemezliğini ifade etmektedir. Teorem kaotik, dinamik ve doğrusal olmayan sistemlerin başlangıç parametrelerine göre değişen(duyarlı), yüksek hassasiyetli

sonular rettiđi ve kk deđiřikliklerin sistem performansında byk deđiřimler yol aabileceđi prensibine dayanır. Kaos sistemleri rastgelelik (randomness) iermektedir ancak rastgele parametrelerle alıřmazlar. Kaos sistemleri ayrıca belirli bir tekrara dřmemeyi garanti ettiđinden ergodiktirler. Yani kaos sistemlerinde rastgele dizilerde olabileceđi gibi aynı sayıdan tekrar retilmesi mmkn deđildir. İřte bu zellik sebebiyle optimizasyon problemlerinde zm uzayı taranırken yerel optimumlara takılmamak amacıyla kaos teoreminden yararlanılarak geliřtirilen kaotik haritalar kullanılabilir. Kaotik haritalar tanımlı denklem ile belirli bir bařlangı parametresi ile istenilen byklkte oluřturulur, ardından ihtiya duyulan pozisyonda (rastgele sayı kullanılması gereken durumlarda) retilen kaotik diziden sıradaki sayı alınarak kullanılır. Metasezgisel poplasyon algoritmalarında bu haritalar sıklıkla kullanılmaktadır, (Alatas vd., 2009), (Muhammad & zkaynak, 2020),(Zhang vd., 2009).

### **3.1.1. Kaotik haritalar**

Kaos teoreminden hareketle retilen kaotik haritalar, istenilen byklkte retilen, saklama ve retilme maliyeti dřk, stokastik, deterministik ve dođrusal olmayan sayı dizilerinden oluřmaktadır. Bu haritalar vasıtasıyla oluřturulan sayılar kendi iinde tekrara dřmeden ve belirli bir sayı aralıđına sıkıřmadan geniř bir spektruma yayılmıř belirli bir sistematıđi olan dzenli sayılardır. Bu sebeple literatrde metasezgisel algoritmalarda zm uzayında gezinirken kullanılan rastgele sayılar yerine kullanılması, olumlu sonular dođurmuřtur. Literatrde farklı amalara hizmet eden ve farklı yntemlerle retilen ok fazla kaotik harita olsa da tez alıřmasında en sık kullanılan 5 adet kaotik harita test edilmiřtir (Turkoglu vd., 2023). Bu haritalar Logistic, Piecewise, Singer, Sinusoidal ve Tent Kaotik Haritalarıdır (Demir vd., 2020) (Gandomi vd., 2013), (Saremi vd., 2014), (Talatahari vd., 2012), (Y. Hu vd., 2014), (Wang vd., 2014). İlgili haritalar tez alıřmasında kullanılırken tm bařlangı kořulları eřitlenmiř ve 0.7 olarak sabitlenmifitir. Seilen kaotik haritalara ait matematiksel denklem ifadesi Tablo 3.1’de grlmektedir.



**Tablo 3.1.** Seçilen kaotik haritalara ait matematiksel gösterim.

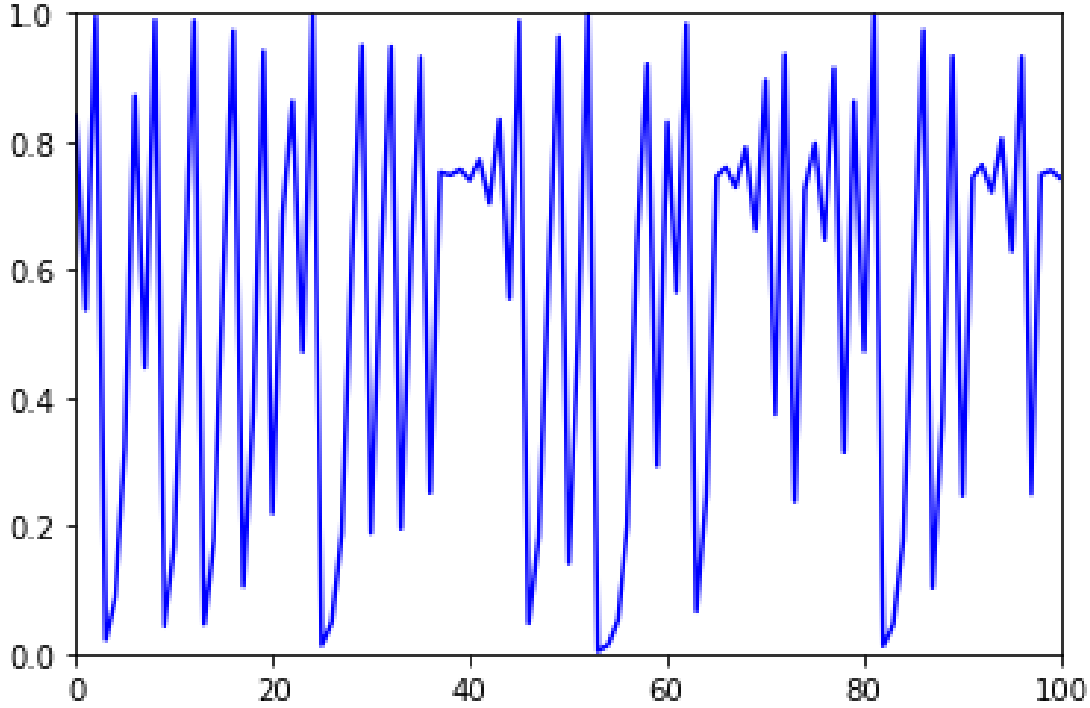
Harita Adı	Denklem
C1= Logistic Harita	$c_{i+1} = w c_i (1 - c_i) \quad (0,1)$
C2= Tent Harita	$c_{i+1} = \begin{cases} c_i / 0.7, & c_i < 0 \\ \left(\frac{10}{3}\right) (1 - c_i), & c_i \geq 0 \end{cases} \quad (0,1)$
C3= Piecwise Harita	$c_{i+1} = \begin{cases} \frac{c_i}{p}, & 0 < c_i < p \\ (c_i - p)/(0.5 - p), & p < c_i < 0.5 \\ (1 - p - c_i)/(0.5 - p), & 0.5 < c_i < 1 - p \\ (1 - c_i)/p, & 1 - p < c_i < 1 \end{cases},$ $p = 0.2 \quad (0,1)$
C4= Singer Harita	$c_{i+1} = \mu (7.86 c_i - 23.31 c_i^2 + 28.75 c_i^3 - 13.302875 c_i^4),$ $\mu = 1.07 \quad (0,1)$
C5= Sinusoidal Harita	$c_{i+1} = w c_i^2 \sin(\pi c_i), \quad w = 2.3 \quad (0,1)$

### 3.1.1.1. Logistic kaotik haritası

k sabiti iterasyon sayısını,  $x_k$  k.kaotik sayıyı,  $\alpha$  değeri ise harita sabiti olmak üzere Logistic Kaotik Haritası'na ait denklem 3.1'de gösterilmektedir. İlgili denklemin kaotik sayı üretiminde kullanılabilmesi için  $\alpha$  değeri [3.57, 4] aralığında bir değer olmalıdır.

$$x_{k+1} = \alpha x_k (1 - x_k) \quad (3.1)$$

Başlangıç parametresi  $X_1 = 0,7$  ve kaotik sabiti  $\alpha = 4$  olmak üzere Lojistik Kaotik Harita denklemi kullanılarak üretilen 100 sayının spektrumu Şekil 3.1'de gösterilmiştir.



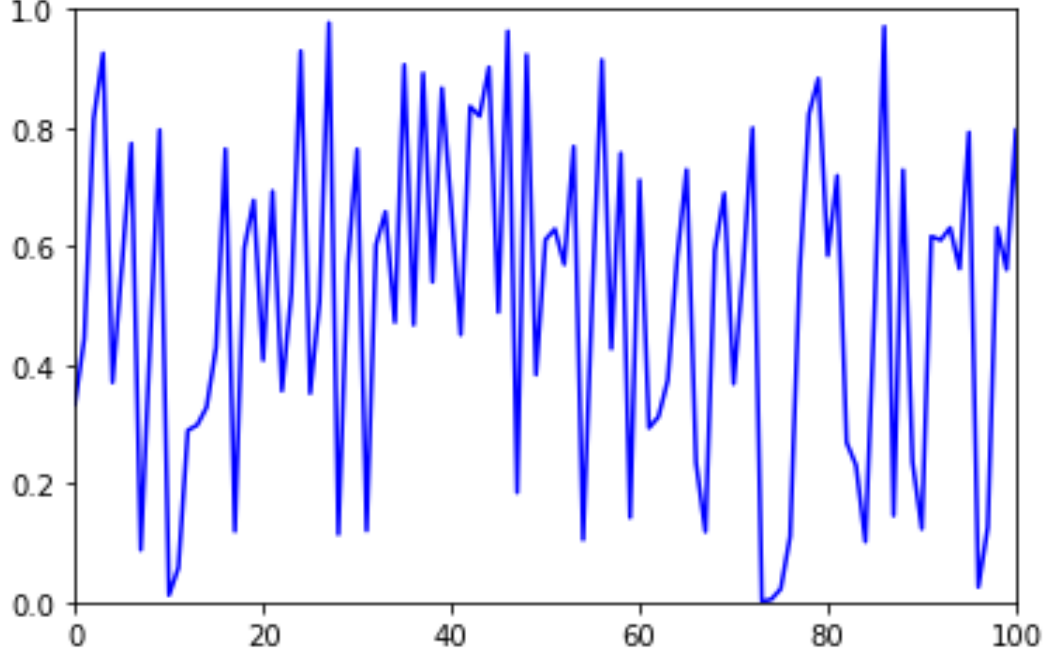
Şekil 3.1. Logistic kaotik haritası yayılım spektrumu.

### 3.1.1.2. Piecewise kaotik haritası

$k$  sabiti iterasyon sayısını,  $x_k$   $k$ .kaotik sayıyı,  $P$  değeri ise harita sabiti olmak üzere Piecewise Kaotik Haritasına ait denklem 3.2'de gösterilmektedir.

$$x_{k+1} = \begin{cases} \frac{x_k}{p}, & 0 < x_k < p \frac{(x_k - p)}{(0.5 - p)}, \\ p < x_k < 0.5 \frac{(1 - p - x_k)}{(0.5 - p)}, \\ 0.5 < x_k < 1 - p \frac{(1 - x_k)}{p}, \\ 1 - p < x_k < 1 \end{cases} \quad (3.2)$$

Başlangıç parametresi  $X_1 = 0,7$  ve kaotik sabiti  $p = 0.2$  olmak üzere Piecewise Kaotik Harita denklemi kullanılarak üretilen 100 sayının spektrumu Şekil 3.2’de gösterilmiştir.

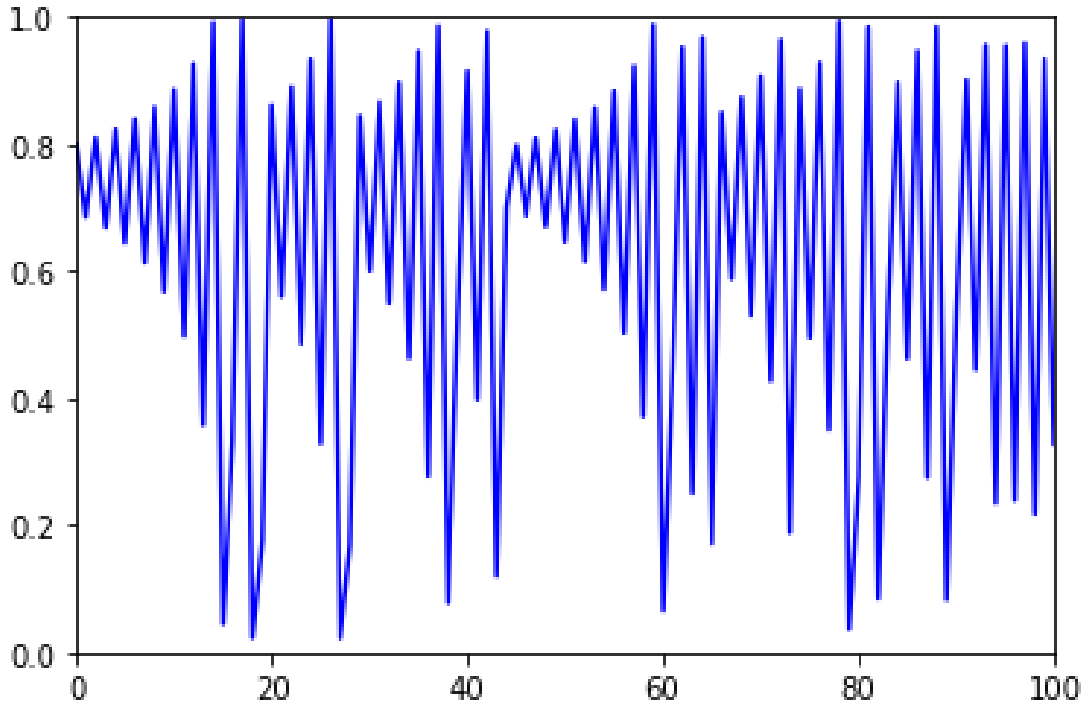


Şekil 3.2. Piecewise kaotik haritası yayılım spektrumu.

### 3.1.1.3. Singer kaotik haritası

$k$  sabiti iterasyon sayısını,  $x_k$  k.kaotik sayıyı,  $\mu$  değeri ise harita sabiti olmak üzere Singer Kaotik Haritasına ait denklem 3.3’te gösterilmektedir. Başlangıç parametresi  $X_1 = 0,7$  ve kaotik sabiti  $\mu = 1.07$  olmak üzere Singer Kaotik Harita denklemi kullanılarak üretilen 100 sayının spektrumu Şekil 3.3’ te gösterilmiştir.

$$x_{k+1} = \mu (7.86 x_k - 23.31 x_k^2 + 28.75 x_k^3 - 13.302875 x_k^4) \quad (3.3)$$



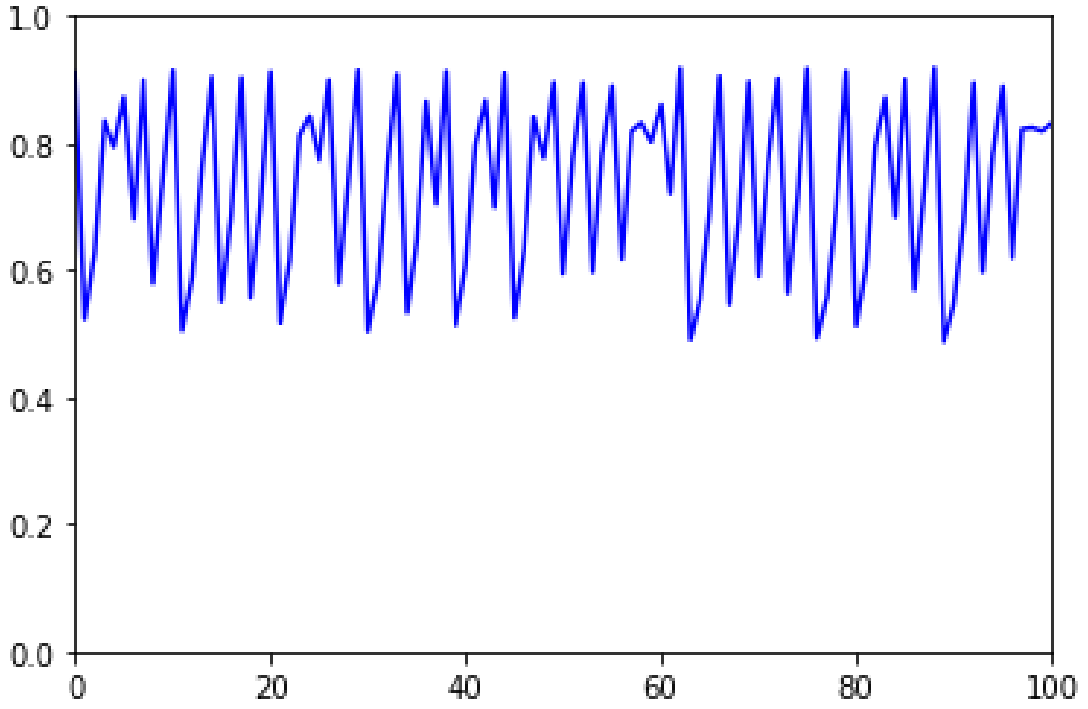
Şekil 3.3. Singer kaotik haritası yayılım spektrumu.

#### 3.1.1.4. Sinusoidal kaotik haritası

$w$  sabiti iterasyon sayısını,  $x_k$  k.kaotik sayıyı,  $\alpha$  değeri ise harita sabiti olmak üzere Sinusoidal Kaotik Haritasına ait denklem 3.4 'te gösterilmektedir.

$$x_{k+1} = wx_k^2 \sin(\pi x_k) , \quad w = 2.3 \quad (3.4)$$

Başlangıç parametresi  $X_1 = 0,7$  ve kaotik sabiti  $w = 2.3$  olmak üzere Sinusoidal Kaotik Harita denklemi kullanılarak üretilen 100 sayının spektrumu Şekil 3.4'te gösterilmiştir.



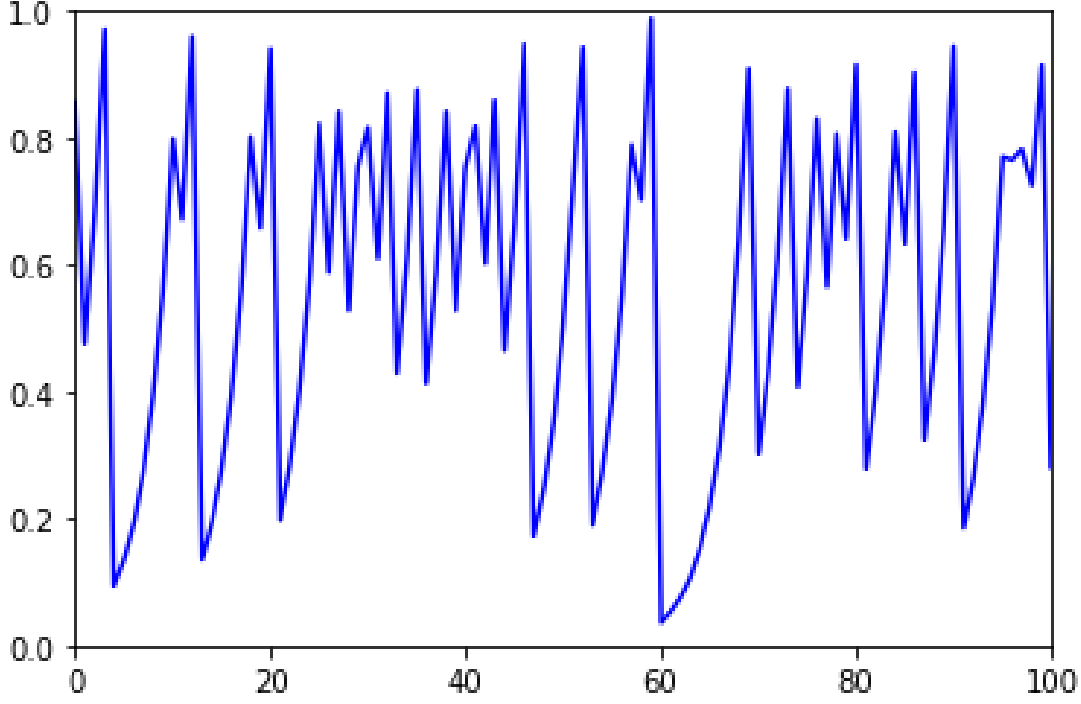
Şekil 3.4. Sinusoidal kaotik haritası yayılım spektrumu.

### 3.1.1.5. Tent kaotik haritası

$k$  sabiti iterasyon sayısını,  $x_k$   $k$ . kaotik sayıyı,  $\alpha$  değeri ise harita sabiti olmak üzere Tent Kaotik Haritası'na ait denklem 3.5'te gösterilmektedir.

$$x_{k+1} = \begin{cases} x_k / 0.7, & x_k < 0 \\ \left(\frac{10}{3}\right) (1 - x_k), & x_k \geq 0 \end{cases} \quad (3.5)$$

Başlangıç parametresi  $X_1 = 0,7$  olmak üzere Tent Kaotik Harita denklemi kullanılarak üretilen 100 sayının spektrumu Şekil 3.5'te gösterilmiştir.



Şekil 3.5. Tent kaotik haritası yayılım spektrumu.

### 3.2. Sınıflandırma ve Kümeleme Algoritmaları

Sınıflandırma ve kümeleme algoritmaları, amaçları veriyi sade bir hale getirerek kullanıcının daha anlaşılır şekilde kullanmasını sağlayan, veri madenciliği ve makine öğrenmesi süreçlerinde kullanılan önemli birer araçtır. Kümeleme, veri noktalarını gruplara ayıştırmak için kullanılan bir yöntemdir. Bu gruplar, benzer özelliklere sahip veri noktalarını içerebilir. Kümeleme yöntemleri arasında en yaygın olanlar arasında K-Means, Hierarchical Clustering ve DBSCAN bulunmaktadır. Kümeleme, benzer veri noktaları gruplarını belirlemek amacıyla işlem yaparken, sınıflandırma, kategoriler veya sınıflar olarak adlandırılan, hali hazırda gruplara bölünmüş örnek veri kümesinin yapısını öğrenme işlemi yapmaktadır. Bu algoritmalar, verileri belirli kategorilere ayırmak için kullanılır. Örneğin, bir sınıflandırma algoritması, e-postaları "spam" ve "ham" olarak sınıflandırabilir. Sınıflandırma algoritmaları önceden bilinmeyen bir veya daha fazla görülmemiş veri örneğinin grup tanımlayıcılarını (veya sınıf etiketlerini) tahmin etmek için kullanılır.

Sınıflandırma problemi girdileri sınıflara ayrılmış örnek veri setlerinden oluşur. Buna eğitim verileri ve bu sınıfların grup tanımlayıcılarına sınıf etiketleri denilmektedir. Çoğu durumda, sınıf etiketlerinin, bir veri nesnelere grubu gibi net bir anlamsal yorumu

vardır. Eğitim amaçlı oluşturulan ve öğrenme sağlayarak tahminde bulunan modele eğitim modeli denir. Bu nedenle sınıflandırma, denetimli öğrenme olarak adlandırılır çünkü bir öğretmenin öğrencilerini belirli bir hedefe doğru yönlendirmesi gibi, grupların yapısını öğrenmek için örnek bir veri seti kullanılır. Eğitim modelinin daha önce hiç karşılaşmadığı veri noktaları, eğitilmiş model üzerinde test edildiğinden bu veri seti test veri seti olarak adlandırılır. Bir sınıflandırma modeli tarafından öğrenilen gruplar, kümelemede olduğu gibi, genellikle özellik değişkenlerinin benzerlik yapısıyla ilişkili olabilirken, bu durum zorundalık içermemektedir. Sınıflandırmada, örnek eğitim verileri, grupların nasıl tanımlandığına dair rehberlik sağlamada çok önemlidir. Bu nedenle, sınıflandırma problemi sezgisel olarak her biri bir sınıf etiketiyle ilişkili bir dizi eğitim veri noktası verildiğinde, daha önce görülmemiş bir veya daha fazla test örneğinin sınıf etiketini belirlenmesidir.

Sınıflandırma algoritmaları genel hatları ile iki aşamadan oluşur:

Eğitim aşaması: Eğitim için ayrılan örneklerden oluşan eğitim modeli sayesinde veri seti üzerindeki etiketli grupların matematiksel olarak modellenmesi aşamasıdır.

Test aşaması: Eğitim modelinin daha önce hiç görülmemiş test örneklerine ait sınıf etiketini tanımlamak amacıyla kullanılan modeldir.

Sınıflandırma problemi, kümelemeden daha güçlüdür çünkü kümelemeden farklı olarak, örnek bir veri kümesinden kullanıcı tanımlı bir gruplama kavramını yakalar. Böyle bir yaklaşım, grupların harici uygulamaya özgü kriterlere göre doğal olarak tanımlandığı çok çeşitli problemlere neredeyse doğrudan uygulanabilirliğe sahiptir. Sınıflandırma uygulamaları, örnek yoluyla öğrenme yeteneği nedeniyle çeşitlidir. Birçok farklı sınıflandırma algoritması mevcut olmasına rağmen yaygın olarak kullanılanlar arasında K-En Yakın Komşuluk (k-NN), Destek Vektör Makinesi (SVM), Lojistik Regresyon, Yapay Sinir Ağları, Rastgele Orman, Naive Bayes ve Karar Ağaçları bulunur. Sınıflandırma uygulamaları da algoritmaları gibi, örnek yoluyla öğrenme yeteneği nedeniyle çeşitlidir.

Çalışmada kullanılan sınıflandırma yöntemleri şunlardır:

### **3.2.1. K en yakın komşu (k-NN) algoritması**

k-En Yakın Komşu (k-NN) algoritması yapay zeka, veri madenciliği, makine öğrenmesi, tıbbi bilişim, istatistik, örüntü tanıma, biyoinformatik gibi bir çok alanda kullanılan ve 1967 yılında önerilen bir sınıflandırma algoritmasıdır (Cover & Hart,

1967), (Keller vd., 1985), (Mao vd., 2015). k-En Yakın Komşu (k-NN) algoritması belirlenen bir k sabit değeri kullanılarak verilerin örnek bir veri noktasına en yakın komşuluk uzaklığının hesaplandığı bir sınıflandırma algoritmasıdır. Verileri belirli kategorilere ayırmak ve veriler arasındaki ilişkiyi benzerlik/farklılık yönüyle tespit etmek amacıyla popüler olarak kullanılır (Batista & Silva, 2009).

Algoritmanın öğrenme metodolojisi eğitim setinde bulunan verilere dayalı olarak değişmektedir. Eğitim setindeki özellikler n boyutlu sayısal nitelikler ile tanımlanır. Ardından her örnek n boyutlu uzayda bir noktayı gösterecek şekilde örneklem uzayı temsil edilir. Eğitim setinde gerçekleşen öğrenme sonrası, test seti üzerinden ilk kez karşılaşılan örneğin öğrenilen benzerliğe göre sınıflandırılması veya başka bir değişle var olan verilere olan uzaklığının hesaplanması mantığına dayanmaktadır (Han & Kamber, 2006), (Mitchell, 2007). Uzaklık hesaplaması k-En Yakın Komşu Algoritmasının performansını önemli ölçüde etkilemektedir. Bu amaçla k-En Yakın Komşu algoritmasının uzaklık ölçümünü farklı şekillerde ağırlıklandırarak belirleyen yöntemler geliştirilmiştir (Gärtner vd., 2004). K-en yakın komşuluk algoritmasında Minkowski, Öklid, Manhattan Mesafe Ölçütleri kullanılsa da en yaygın olarak kullanılan Öklid mesafe ölçümüdür.

### 3.2.1.1. Minkowski mesafe ölçümü

Minkowski mesafesi, öklid uzayında tanımlı bir dizidir ve denklem 3.6'da ifade edilmiştir. Belirli sayıda (p) değişkene bağlı uzaklık hesaplaması yapılmak istendiğinde kullanılır (Taşcı & Onan, 2016).

$$\left( \sum_{i=1}^n |x_i - y_i| \right)^{1/p} \quad (3.6)$$

### 3.2.1.2. Öklid mesafe ölçümü

Öklid uzaklığı, pisagor teoreminin uygulaması olarak iki nokta arasındaki doğrusal uzaklığı hesaplar ve denklem 3.7'de ifade edilmiştir. En sık kullanılan uzaklık ölçütüdür (Taşcı & Onan, 2016).



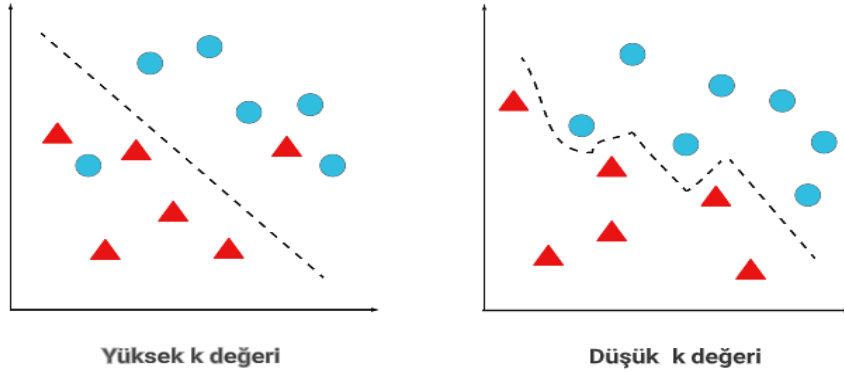
$$\sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.7)$$

### 3.2.1.3. Manhattan mesafe ölçümü

Manhattan uzaklığı,  $n$  boyutlu iki nokta arasındaki uzaklık farkını kullanır ve denklem 3.8'de ifade edilmiştir (Taşcı & Onan, 2016).

$$\sum_{i=1}^n [x_i - y_i] \quad (3.8)$$

k-NN algoritmasının uygulamasında k değeri k sayıdaki komşuların yakınlığı ve uygunluğu ile sınıflandırma yapılmasını sağlamaktadır. K değeri performansı önemli ölçüde etkilediğinden en uygun değer olarak seçilmesi gerekmektedir. Birbirine en yakın iki komşu arasında karmaşıklığı azaltmak amacıyla genellikle tek sayı olarak seçilen k değeri çok yüksek değer olarak seçildiğinde modelin özellikle gürültülü verideki performansını düşürmektedir. Bu durum modelin eğitim sürecinde sapmalara sebebiyet verir.



Şekil 3.6. k değeri seçimi.

Benzer şekilde k değerinin çok düşük seçilmesi sonucunda da veriler üzerinde modele bağlı olarak aşırı öğrenme gerçekleşir. Bu durumun bir sonucu olarak modelin hata oranı beklenilenden çok daha yüksek olacaktır. İlgili durum Şekil 3.6'da görülebilmektedir.

k-NN sınıflandırma algoritmasının kolay uygulanabilirliği, ölçeklenebilirliği, analitik olarak gözlemlenebilmesi, paralel olarak gerçekleştirilebilmesi, gürültülü verilere karşı dirençli olması gibi avantajlarının yanı sıra, performansını gerçekleştirmek için yüksek bellek miktarına ihtiyaç duyması ve veri seti boyutu arttıkça hesapsal maliyetin artması, performansının k değeri ve belirlenen uzaklık ölçüsüne göre değişmesi gibi çeşitli dezavantajları bulunmaktadır (Bhatia, 2010),(H. Liu & Zhang, 2012).

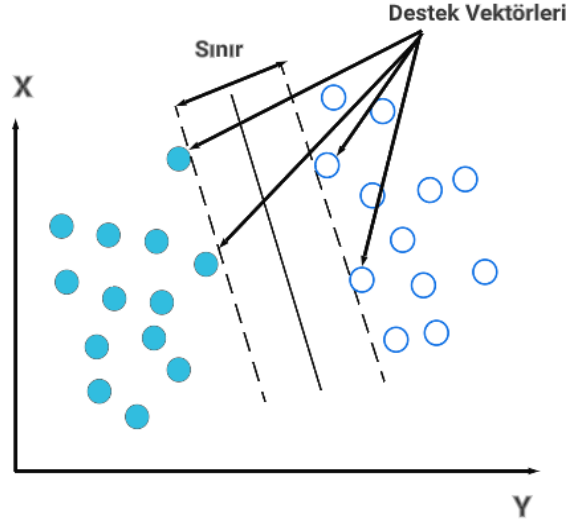
### **3.2.2. Destek vektör makineleri (SVM)**

Destek Vektör Makineleri (SVM) sınıflandırma, regresyon problemi, örüntü tanıma problemlerinin çözümü için karmaşık ve çok boyutlu veri kümelerinde etkili Vapnik tarafından geliştirilen denetimli bir makine öğrenimi yöntemidir (Cortes & Vapnik, 1995). Önceleri iki sınıflı doğrusal problemlerin sınıflandırılması için tasarlanırsa da, sonrasında çok sınıflı ve doğrusal olmayan verilerin sınıflandırılmasında kullanılmıştır. SVM, veri noktalarının iki veya daha fazla sınıfa ayrılmasını sağlamak için en uygun hiperdüzlemi arar. Bu hiperdüzlem, veri noktalarını en iyi şekilde ayıran veya en az hata oluşturacak şekilde seçilir (Cortes & Vapnik, 1995). SVM'ler, veri noktalarının etrafındaki, veri noktaları arasındaki aralığı öteleme veya genişletme yoluyla maksimize etmeyi amaçlar. Bu aralık, sınıflandırma için kullanılan hiperdüzlemi belirler. Verilerin doğrusal olarak ayrılabilir olduğu durumlara, iki sınıfa ait veri noktalarını temiz bir şekilde ayıran bir doğrusal hiper düzlem oluşturmak mümkündür. Ancak çok sınıflı veri kümelerinde de hızlı eğitim tekniklerinin geliştirilmesini sağlamıştır (Amiri & Armano, 2013). Genellikle çok boyutlu veri kümelerinde kullanılsalar da düşük boyutlu veri kümelerinde de kullanılabilir. SVM'ler, çok sayıda parametre ayarlaması gerektirir. Bu parametreler arasındaki ilişki ve etkisi, SVM'nin performansını etkileyebilir. Bu nedenle, SVM uygulamalarında parametre seçimi önemlidir.

Destek vektör makineleri doğrusal olarak ayrılabilen ve doğrusal olarak ayrılamayan olmak üzere temel olarak ikiye ayrılır:

### 3.2.2.1. Doğrusal olarak ayrılabilen destek vektör makineleri

Doğrusal olarak ayrılabilen SVM, veri noktalarının iki sınıf arasında doğrusal bir hiperdüzlem tarafından ayrılacağı durumlarda kullanılır. Yani girdi örneklerini ayırabilen sınır belirlenerek veri kümesinin ayrışması mantığına dayanmaktadır. Bu sınırın iki sınıfın belirlenen çizgisinin en yakınında bir nokta olması amaçlanır. Bu nokta sınır çizgisine en yakın olduğundan destek noktası olarak adlandırılır. Veri setinin doğrusal olarak ayrılma durumuna ilişkin destek vektör makinelerinin geometrik yapısı Şekil 3.7’de gösterilmiştir. Hiper düzlemin sol tarafında bir sınıf; sağ tarafında ise diğer sınıf bulunmaktadır. Optimum hiper düzlem sınıfların sınırları arasında bulunma şartını sağlamak zorundadır. Bu şekilde doğrusal olarak dağılan veri kümeleri SVM yardımıyla sınıflandırılmış olur.

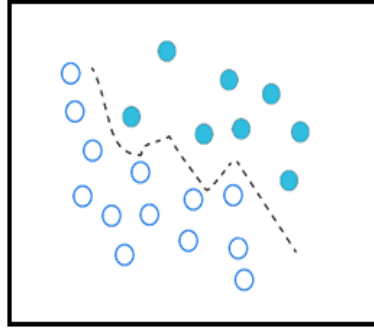


Şekil 3.7. Doğrusal olarak ayrılabilen veri setleri için hiper-düzlemin belirlenmesi (Kavzoğlu & Çölkesen, 2010).

### 3.2.2.2. Doğrusal olarak ayrılamayan destek vektör makineleri

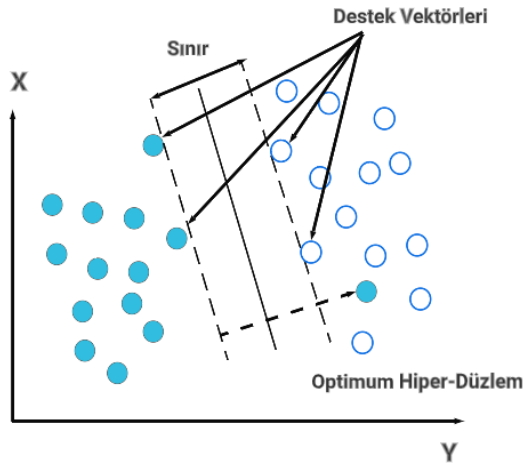
Gerçek dünya problemleri genellikle doğrusal olmadığından probleme ait verilerin keskin bir şekilde hiper düzlem üzerinde ayrılamaması durumunda doğrusal olarak ayrılamayan SVM kullanılır. Bu sebeple bu ayırım ancak tahmini bir şekilde yapılabilir. Ancak böyle bir tahminin yapılması da oldukça güçtür. Bu sebeple uygulamada doğrusal olmayan bir haritalama kullanılarak SVM’lerin yüksek boyutlu uzayda optimum bir ayırma işlemi hiper düzlem oluşturulması ile sağlanır (Cortes & Vapnik, 1995).

Veri noktalarının doğrusal olarak ayrılamadığı durumlarda veri noktalarını daha yüksek boyutlu bir uzayda optimum ayırma ve doğrusal olarak ayrılabilir şekilde düzenlenmesi kernel fonksiyonları ile sağlanabilir. Böylece SVM'nin performansı da artırılmış olmaktadır (Maria & Yassine, 2020). Doğrusal olarak ayrılamayan veri seti Şekil 3.8'de gösterilmiştir.



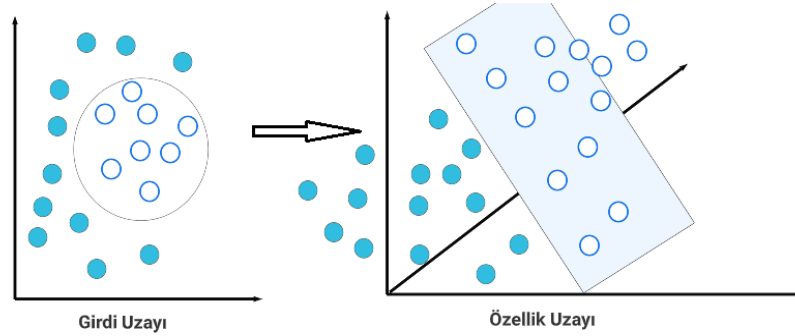
**Şekil 3.8.** Doğrusal olarak ayrılamayan veri seti(Kavzoğlu & Çölkesen, 2010).

Doğrusal olarak ayrılamayan veri setleri için hiper-düzlemin belirlenmesi Şekil 3.9'da gösterilmiştir.



**Şekil 3.9.** Doğrusal olarak ayrılamayan verilerin hiper-düzlemin belirlenmesi (Kavzoğlu & Çölkesen, 2010).

SVM'ler, kernel teknolojisi kullanarak düşük boyutlu verileri yüksek boyutlu hiperdüzlemlere dönüştürebilir. Bu işlem, veri noktaları arasındaki aralığı maksimize etmeyi kolaylaştırır ve sınıflandırma işlemi için daha iyi sonuçlar verir. Kernel fonksiyonları verileri daha yüksek boyutlu hiperdüzlemlere dönüştürerek, lineer olmayan sınırların tahmin edilmesini sağlar. Kernel fonksiyonu ile verinin daha yüksek bir boyuta dönüştürülmesi Şekil 3.10'da sembolize edilmiştir. Ancak her kernel fonksiyonu Destek Vektör Makinelerinde kullanımı uygun olmamaktadır. Dolayısıyla, SVM'nin uygulanmasında kernel fonksiyonunun seçimi kritik rol oynamaktadır. Ayrıca literatürde birçok kernel fonksiyonu tanımlanmıştır. SVM için en sık kullanılan kernel fonksiyonları radyal tabanlı, polinomial, lineer ve sigmoid kernelleridir (Osuna, 1998).



Şekil 3.10. Kernel fonksiyonu ile verinin daha yüksek boyuta dönüştürülmesi.

### 3.3. Öznitelik Seçimi

Öznitelik seçimi, veriye ait niteliklerden hedef değişkenle en alakalı olanların bir alt küme olarak tespit edilmesi tekniğidir. Makine öğrenme algoritmaları çok sayıda veri ile çalıştığında sistem bellek kullanımını arttırdığından eğitim süreci yavaşlayarak, modelin gelişmesi ve tahmin performansı düşebilmektedir. Bu nedenle hedef değişkenle ilişkisi olmayan veriler tahmin belirsizliğini artırırken, modelin genel etkinliğini de azaltabilmektedir. Veri kümelerindeki girdi değişkenlerinden alakasız ve gereksiz olanların kullanılmaması yani veri kümesinden kaldırılması ile bu sorun giderilebilir, öğrenme algoritmasının performansı artırılabilir. Hem modelin hesaplama maliyetini hem de performansını iyileştirmek için girdi değişkenlerinin sayısını azaltmak arzu edilmektedir.

Öznelik seçimi, modele ait daha az girdi değişkeni hedeflediğinden boyut azaltma teknikleriyle benzerlik göstermektedir. Aradaki fark, öznelik seçiminin veri kümesinde tutulacak veya veri kümesinden kaldırılacak özellikleri seçmesi, oysa boyut azaltmanın tamamen yeni girdi özellikleriyle sonuçlanan bir veri projeksiyonu oluşturmasıdır. Bu nedenle, boyut azaltma öznelik seçimi yerine özellik seçimi işlemine bir alternatiftir.

Öznelik seçim teknikleri genellikle hedef değişkeni kullananlar (denetimli) ve kullanmayanlar (denetimsiz) olmak üzere ikiye ayrılır. Öznelik seçim sürecinde ilk yapılması gereken ayırım denetimli ve denetimsiz seçimden hangisinin yapılacağıdır. Denetimsiz öznelik seçim teknikleri, hedef değişkeni kullanmadan işlem yapar ve sonuçlar göz ardı edilir. Örneğin, veri temizleme işlemleri denetimsiz öznelik seçimine örnektir. Denetimli öznelik seçim teknikleri ise ilgisiz değişkenleri kaldıran yöntemler gibidir ancak bu işlemi hedef değişkeni kullanarak yapar (Brownlee, 2020).

Denetimli öznelik seçim yöntemleri Filtre, Sarmalayıcı ve Hibrit Model olmak üzere 3 ana sınıfta değerlendirilmektedir.

### **3.3.1. Filtreleme model**

Filtre tabanlı öznelik seçim yöntemleri, istatistiksel ölçümler kullanmaktadır. Her bir girdi değişkeni ile hedef değişkeni arasındaki ilişkinin değerlendirildiği ve aralarındaki ilişkiye ait güçlü bağ olanların tespitinin yapıldığı ve ardından bu girdilerin seçildiği yöntemlere filtre tabanlı öznelik seçimi denir. Değerlendirilecek veri türüne bağlı olarak istatistiksel yöntem seçilir. Doğru seçim yapıldığında filtre modelli yöntemler hızlı ve etkili sonuçlar üretebilmektedir. Bu sebeple makine öğrenmesi sistemlerinde filtre tabanlı öznelik seçimi gerçekleştirilirken kullanılan veri kümesine uygun istatistiksel ölçü seçimi yapmak zor olabilmektedir. Filtreleme yöntemleri özellikleri tek başlarına değerlendirir, birbirleri ile olan olası ilişkisini göz önüne almaz. Model özelliklerin birbirinden bağımsız olduğu varsayımına dayalı çalıştığından, hesaplama karmaşıklığı daha düşük olmasına rağmen, birçok uygulama için modelin performansı iyi değildir. Filtreleme modelli öznelik seçim işleminde özellikler bazı istatistiksel yöntemlere göre sıralanarak içlerinden en iyi N tane özellik belirlenerek öznelik kümesi olarak seçilir. Veri kümesindeki hedef sınıfa duyarlı özelliklere filtre modeli uygulandığında bir grup özellik aynı anda değerlendirildiğinden gereksiz verilerin tespiti daha kolay olmaktadır.

Ki kare istatistiği, t istatistiği, bilgi kazancı, minimum artıklık ve maksimum alaka, relief, ortak bilgi, korelasyon skorlama gibi pek çok istatistiksel filtreleme yöntemi mevcuttur (Sánchez-Marño vd., 2007a). Örneğin, iki özellik değişkeninin olduğu durumlarda yapılan işlemi düşünürsek bu özellikler birbiriyle mükemmel bir şekilde ilişkilidir ve bu nedenle her biri kullanılarak tahmin yapılabilir. Buna benzer durumlarda özelliklerden birini kullanmak mantıklıdır. Bununla birlikte, bu tür yöntemler genellikle hesaplama karmaşıklığı açısından maliyetlidir çünkü üzerinde bir arama yapılacak boyut alt kümesi olasılığı kadar özellik alt kümesi vardır. Çoğu uygulamada öznitelik seçme yöntemleri nitelikleri birbirinden bağımsız olarak değerlendirilerek ayırt edici olanları seçmeye çalışır. Bu sebeple çoğu öznitelik seçim yöntemi bağımsız sınıflandırıcıların kullanıldığı boyut indirgeme yöntemi olarak da değerlendirilebilir.

### 3.3.2. Sarmalayıcı model

Filtre yöntemlerinde kullanılanın aksine sarmalayıcı yöntemlerde, özellikler tek başlarına değil birbirlerine bağlı alt gruplar halinde değerlendirilmektedir. Sarmalayıcı yöntemler, model performansını en üst düzeye çıkaran en uygun kombinasyonu bulmak amacıyla özellikleri çeşitli kombinasyonlarla kullanarak birden çok kez modeli değerlendirir. Önceden belirlenen belirli bir kural kullanarak veya rastgele seçilerek alt gruplar oluşturulmaktadır. Elde edilen alt grup/kümelerin kalitesi literatürce iyi bilinen kümeleme ya da sınıflandırma algoritmaları ile değerlendirilmektedir. Sarmalayıcı yöntem filtreleme yönteminden daha iyi sonuçlar üretmesine rağmen hesapsal maliyetleri özellik sayısına göre katlanarak arttığından daha yüksektir (Kohavi & John, 1997). Özelliklerin birbirinden bağımsız olmadığı ve sonuca etki ettiği varsayımı ile hareket edildiğinden, özelliklerin aralarındaki ilişki göz önüne alınarak alt küme seçimi yapılması kısmen de olsa hesapsal karmaşıklığı azalttığı gibi, performansı arttırarak daha iyi sonuçlar üretilmesine sebep olur.

Sarmalayıcı yöntemlerde özellikle büyük veri kümelerinde tüm olasılıklar göz önünde bulundurularak alt küme oluşturma işlemi mümkün olmadığından, sezgisel yöntemler (doğadan esinlenmeli algoritmalar) alt küme seçim aracı olarak sıklıkla kullanılmaktadır. Bu algoritmalar çözüm uzayında, tüm özellikleri test etmese de, sezgisel bir arama yapıp yerel ve küresel arama stratejisi ile özellik uzayının keşfini sınırlandırarak, daha makul sürelerde daha efektif sonuçların elde edilmesini sağlamaktadır (Got vd., 2021). Performansları nedeniyle, Genetik Algoritma (GA),

Karınca Kolonisi Optimizasyonu (KKO), Parçacık Sürü Optimizasyonu (PSO), Diferansiyel Evrim (DE) Algoritması gibi birçok sezgisel algoritma, nitelik seçimi problemine en sık uygulanan yöntemlerdendir.

### **3.3.3. Hibrid/Embedded model**

Hibrit yaklaşımlar da ise, hem sarmalayıcı hem de filtre yönteminin avantajlarından yararlanmak için iki yöntem birleştirilerek uygulanmaktadır. Bu yöntemde, özellik boyutunu azaltmak için öncelikle filtreleme yöntemi ile ön işlem uygulanır, ardından kalan özellikler arasından en iyi alt kümeyi bulabilmek için bir sarmalayıcı yöntem kullanılmaktadır. Kısaca bu yöntemde iki modelin uygulama açısından iyi özellikleri birleştirilerek eş zamanlı şekilde kullanılmaktadır (Tran vd., 2016). Özellik alt küme seçimi, bir arama metodu ve aday özellik kümesinin sınıflandırmaya katkısını puanlayan bir değerlendirme stratejisinden oluşmaktadır. Kullanılan alt küme arama metotları;

1. Tam arama,
2. Rastgele arama,
3. Sıralı arama,
4. Artan arama ve
5. Meta sezgisel arama olmak üzere beşe ayrılır (Sánchez-Marono vd., 2007b).

Arama işlemi neticesinde iyi sonuç veren özellikler oluşturulan alt küme eklenirken, kötü özellikler oluşturulan kümeden çıkarılmaktadır. Sınıfı bilinen veri kümeleri üzerinden oluşturulan alt kümeler kullanılarak eğitim süreci ile öğrenme gerçekleştirilir.

## **3.4. Optimizasyon Algoritmaları**

Optimizasyon bir çok ticari faaliyette kullanıldığı gibi mühendislik problemlerinde ve çeşitli endüstriyel tasarımlar da da oldukça fazla kullanılmaktadır. Enerji tüketimini azaltmak, her hangi bir süreçteki maliyetleri azaltmak, çıktıyı, performansı ve verimliliği en üst düzeye çıkarmak gibi sayılamayacak kadar çok çeşitli amaçlar için optimizasyon yapılabilmektedir. Gelişen teknoloji ile tasarım problemlerinden iş akış süreçlerine, internet ve bankacılık işlemlerinden tatil planlamasına birçok sistemde optimizasyona ihtiyaç olduğu görülmektedir. Gerçek dünya uygulamalarında



kaynaklar, zaman ve para her zaman sınırlı olduğundan, bu değerli kaynakları çeşitli kısıtlamalar altında en iyi şekilde kullanmak için çözümler bulunmalıdır. Kısaca herhangi bir sistemde sistemi etkileyen tüm unsunlar için optimizasyon düşünülebilir.

Çoğu gerçek dünya uygulamasına ait problemler doğrusal olmadığından, sıradan yöntemler ve matematiksel araçlar kullanılarak çözüm bulunamaz. Bu gibi durumlarda çözüm için gelişmiş optimizasyon araçları gereklidir. Günümüzde bilgisayar simülasyonları sayesinde çeşitli algoritmalar kolaylıkla problemlere uygulanabilmektedir. Herhangi bir simülasyonun ardında hesaplama modeli oluşturan bazı algoritmalar bulunmaktadır. Bu tür algoritmalarda temel bileşen ve etkileşimli kullanılan araçlar algoritmanın performans ve verimliliğini önemli ölçüde etkiler.

### **3.5. Metasezgisel Optimizasyon Algoritmaları**

Stokastik algoritmalar için, genel olarak buluşsal ve sezgisel olmak üzere iki tür algoritma bulunmaktadır. Buluşsal kelimesi süreç içerisinde aranan varlığı bulmak ya da deneyerek keşfetmek manasında kullanılmaktadır. Optimizasyon problemleri için bu kavram, arama işleminde kaliteli çözümlerin makul bir sürede bulunmasına rağmen optimal çözümün bulunan çözüm olduğuna dair bir garanti olmadığını ifade eder. Bu algoritmaların her zaman olmasa da çoğu zaman doğru şekilde çalışması beklenebilir. Bu durumda buluşsal algoritmalar mutlaka en iyi çözümleri bulmayı garanti edemese de, kolayca ulaşılabilen iyi çözümleri bulacağını garanti eder. Sezgisel yöntem, deneme yanılma yoluyla, karmaşık bir soruna makul ölçüde pratik bir zamanda kabul edilebilir çözümler üretmenin bir yoludur. İlgili problemin karmaşıklığı, kabul edilebilir bir zaman ölçeğinde iyi ve uygulanabilir çözümler bulmalıdır ancak en iyi çözümlerin bulunabileceğinin garantisi yoktur (Yang, 2020).

Sezgisel algoritmalar ayrıca metasezgisel olarak da nitelendirilir, çünkü meta "ötesi" veya "daha yüksek seviye" anlamına gelir ve bu algoritmalar genellikle basit buluşsal yöntemlerden daha iyi performans gösterir. Ayrıca, tüm sezgisel algoritmalar temel prensip olarak rastgelelik kullanırken, yerel arama farklılıkları üzerine kurgulanmıştır. Literatürde buluşsal ve metasezgisel yöntemler için üzerinde anlaşmaya varılmış tanımlar tam olarak bulunmadığından bazen sezgisel ve metasezgisel terimler birbirinin yerine kullanılabilir. Buna rağmen çalışmalar incelendiğinde rastgelelik ve yerel arama yöntemi içeren tüm stokastik algoritmalar metasezgisel bir yöntem olarak değerlendirilmektedir. Rastgelelik yerel aramadan uzaklaşmadan küresel ölçekteki

aramaya geçmek için kullanılan iyi bir yoldur. Rastgelelik kullanan neredeyse tüm algoritmalar küresel ölçekteki arama işlemi için de belirli düzeyde işlem yapmış kabul edilebilir.

Herhangi bir metasezgisel algoritmanın iki ana bileşeni, sömürü ve keşif aşamalarıdır. Keşif, arama alanını küresel ölçekte keşfetmek için farklı çözümler üretmek anlamına gelir. Sömürü ise bu bölgede mevcut iyi bir çözümün bulunduğu bilgisinden yararlanarak yerel bir bölgede aramaya odaklanmak anlamına gelir. En iyinin seçimi, çözümlerin optimalliğe yakınsamasını sağlarken, rastgeleleştirme yoluyla çeşitlendirme, çözümlerin yerel optimumda takılı kalmasını önler ve aynı zamanda çözümlerin çeşitliliğini artırır. Bu iki ana bileşenin iyi bir kombinasyonu, genellikle küresel optimalliğin elde edilebilir olmasını sağlayacaktır.

Metasezgisel algoritmalar birçok şekilde sınıflandırılabilir. Bir sınıflandırma ölçeği olarak doğadan esinli popülasyona dayalı veya yörüngeye dayalı algoritmalar kullanılabilir. Örneğin, Genetik Algoritmalar, Parçacık Sürüsü Optimizasyonu (PSO), Ateş Böceği Algoritması (FA) ve Guguklu Arama Algoritmaları popülasyon tabanlıdır. Diğer taraftan yörüngeye dayalı algoritmalarından Simüle Edilmiş Tavlama, tasarım alanı veya arama alanı boyunca parçalı bir tarzda hareket eden tek bir ajan veya çözüm kullanır. Daha iyi bir hamle veya çözüm her zaman kabul edilirken, çok iyi olmayan bir hamle belli bir olasılıkla kabul edilebilir. Adımlar veya hareketler, arama uzayında sıfır olmayan bir olasılıkla bu yörüngeye küresel optimuma ulaşmasıyla bir yörünge çizer.

### **3.6. Doğa Esinli Optimiazasyon Algoritmaları**

Optimizasyon problemlerinin çözümünde kullanılan geleneksel veya klasik yöntemler deterministik olduklarından, gerçek dünya problemlerini çözmekte yetersiz kalmaktadır. Çözüm süresi ve karmaşıklığının da problemin boyutsallığı ile doğru orantılı olarak artması, bilgisayarların gelişmesi ile birlikte sezgisel araştırma yöntemlerinin araştırılmasına ve gelişmesine yol açmıştır (ERDOĞMUŞ, 2016).

Doğada var olan canlı türlerinin yaşamak için gösterdikleri davranışların tamamı kendi içinde optimum bir tasarıma sahiptir. Doğal olarak yiyecek arama, güvenlik ve hayatta kalma davranışları bilim insanları tarafından incelendiğinde doğadan esinlenen algoritmalar geliştirilebilmektedir. 1970 yıllarından itibaren araştırılan bir çok sezgisel algoritma doğadan esinlenerek geliştirilmiştir. Geliştirilen yöntemlerin bazıları

doğadaki süreçleri fiziksel olarak simüle ederken bazıları da canlıların sosyal etkileşimini örneklemiştir. Canlıların kendi koordinasyonlarını kullanarak özellikle yiyecek bulma amaçlı gösterdikleri zeka, sürü zekası olarak adlandırılmaktadır (Yang, 2020).

### **3.6.1. Orijinal at sürüsü algoritması (HOA)**

At Sürü Optimizasyonu Algoritması(HOA), bir sürüdeki atların doğal yaşam ortamlarındaki davranış kalıplarından ilham alan ve sürü halinde hareket etmesini modelleyen, biyo-ilhamlı metasezgisel bir optimizasyon algoritmasıdır (MiarNaeimi vd., 2021b). HOA, bir sürüdeki atların engellerden ve avcılardan kaçınırken besin kaynağına doğru hareket etme eğiliminde olduğu fikrine dayanmaktadır. Algoritma, sanal bir sürüdeki atların hareketini simüle ederek belirli bir sorun için en uygun çözümü bulmak için bu davranışı kullanır. Algoritma, sorunun çözümüne olası bir çözümü temsil etmesi amacıyla ve rastgele oluşturulan bir at sürüsü popülasyonu ile başlar. Atlar başlangıçta rastgele olarak arama alanına yerleştirilir. Algoritma daha sonra her atın uygunluğunu değerlendirir, bu değerlendirme atın çözümünün soruna ne kadar iyi uyduğunu temsil eder. En iyi atlar daha sonra yeni nesli oluşturmak için seçilirken diğerleri elenir. Algoritma daha sonra en uygun çözümü aramaya rehberlik etmek için sürüdeki atların hareketini kullanır. Sürüdeki her at, engellerden ve avcılardan kaçınarak sürüdeki en iyi ata doğru hareket eder.

Atlar farklı yaşlarda farklı davranışlar sergiler. Atların yaşam alanlarındaki davranışları incelendiğinde, genellikle Otlatma(G), Hiyerarşi (H), Sosyallik (S), Taklit (I), Savunma mekanizması (D) ve Dolaşımı (R) davranışlarını sergiledikleri görülmüştür. Bir at ortalama 25-30 yıl yaşar. At yaşları 0-5, 5-10, 10-15 ve 15'den büyük olarak nitelendirilip kategorize edilmiştir. Atlar çeşitli yaş aralıklarına göre belirlenen isimlerle gruplanmıştır.

$\delta$  atları  $\rightarrow$  0-5 yaş aralığındaki atları,

$\gamma$  atları  $\rightarrow$  5-10 yaş aralığındaki atları,

$\beta$  atları  $\rightarrow$  10-15 yaş aralığını

$\alpha$  atları  $\rightarrow$  15 yaşından yaşlı atları sembolize eder.

Algoritma prensip olarak atların yaşamları boyunca sergiledikleri davranış kalıpları ile farklı yaşlardaki sosyal yaşam performanslarını sembolize etmektedir. Bahsedilen 6

davranış biçimi atlarda yaş almayı temsil eder. Sürüdeki atların konumlarına göre uygunluk değeri hesaplanır. Bu değer iyiden kötüye sıralanarak her yineleme başında atların yaşları ve davranışları güncellenir. Orijinal HOA'ya ait sözde kod Şekil 3.11'de gösterilmiştir. Orijinal HOA'nın tüm çalışma adımları ise Şekil 3.12'de gösterilmiştir.

```
1 : Başla
2 : HOA parametrelerini ayarla:  $N, D, T^{max}, X^{max}, X^{min}$ 
3 : HOA popülasyonu oluştur
4 :  $t=1$ 
5 : while ( $iter \leq T^{max}$ ) do
6 :      $F(X_m)$  hesapla  $\forall m = 1, 2, \dots, N$  (Uygunluk değeri hesaplama)
7 :     At Yaşı Tayini :  $YAS = \alpha, \beta, \gamma, \delta$ 
8 :     Hız Güncellemesi Yap (Denklem 3.10, 3.11, 3.12, 3.13)
9 :     Pozisyon Güncellemesi Yap
10:  $t= t+1$ 
11: end while
12: En iyi sonucu döndür
13: Bitir
```

Şekil 3.11. Orijinal HOA sözde kodu.



Şekil 3.12.Orjinal HOA'nın tüm çalışma adımları.

Sonuç olarak, atlar yaşlarına göre sıralandığında sıralanan popülasyondaki atlardan uygunluk değeri iyi olan atların ilk %10'u  $\alpha$  atları olarak seçilir. Sıralamaya göre ardından gelen %20'yisi  $\beta$  atları,  $\gamma$  ve  $\delta$  atları ise kalan atların sırasıyla % 30 ve % 40'ından oluşur. Dolayısıyla algoritma, atların yaşamları boyunca sergiledikleri davranış kalıpları ile farklı yaşlardaki sosyal yaşam performanslarını sembolize eden altı genel hareketten esinlenerek oluşturulmuştur. Atlara her yinelemede uygulanan hareket denklem 3.9'a göredir. Atların yaşlarına göre sergiledikleri davranışlar aşağıda açıklanmıştır.

$$X_{mIter,YAS} = V_{mIter,YAS} + X_{m(Iter-1),YAS} \quad YAS = \alpha, \beta, \gamma, \delta \quad (3.9)$$

### 3.6.1.1. Otlatma

Atlar bitki, ot ve yem vb. gıdaları otlayarak beslenen hayvanlardır. Atlar yaşamları boyunca her yaşta otlanır. Meralarda dolaşarak bir gün içerisinde ortalama 16 - 20



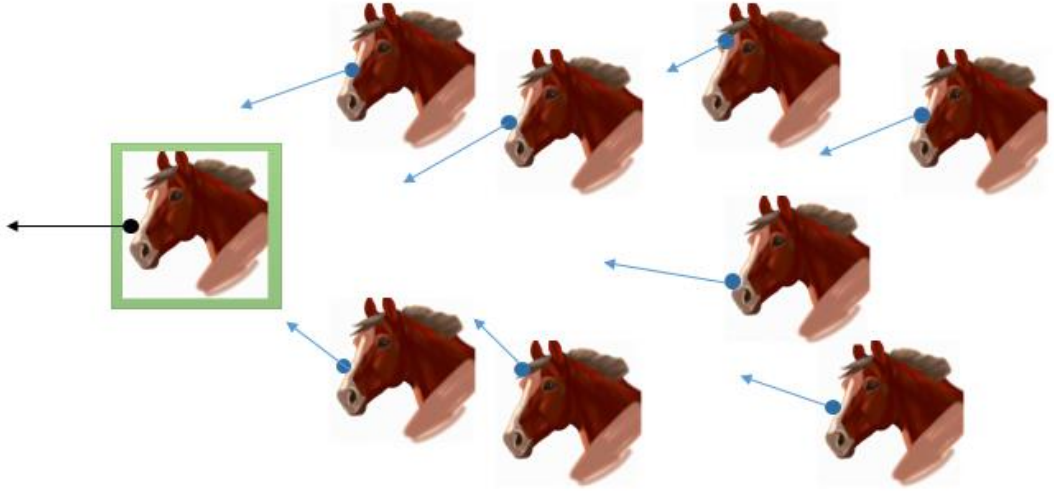
**Şekil 3.13.** Atların otlama davranışı.

saat otlanırlar ve kısa süre dinlenirler. Uzun süren bu otlatma hareketine sürekli yeme hali denir. Atların serbest hareket ederek günün belirli bir kısmını harcadıkları sürekli yeme davranışı Otlatma davranışı olarak ifade edilir. Bu davranış Şekil 3.13'te gösterilmiştir. At her yaşta ve her yöne doğru otlayabilir, otlama yönü için herhangi bir kısıt bulunmamaktadır. Atın otlama hareketi algoritmada lineerliğini azaltacaktır.

Otlama davranışı önerilen algoritmada arama uzayının alt ve üst sınırı kontrol etmek için kullanılmıştır.

### 3.6.1.2. Hiyerarşi

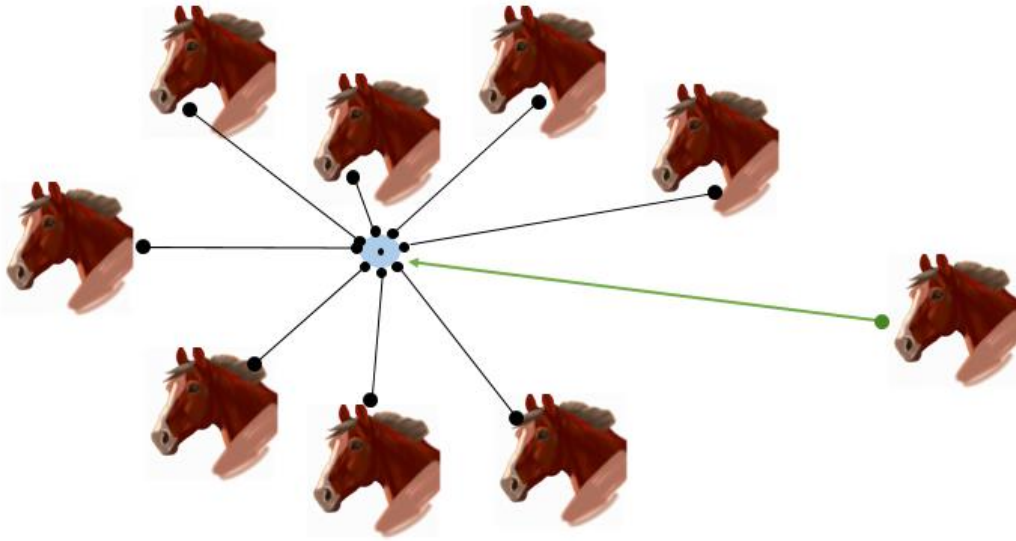
Atlar kendi başlarına özgür değildir. Çoğu zaman insanoğlunun üstlendiği bir liderin peşinden giderek hayatlarını sürdürürler. Sürülerindeki en güçlü ve deneyimli yetişkin at liderlik konumunda olduğundan atlar tarafından takip edilir. Atların bu davranışı Hiyerarşi olarak adlandırılır. Bu davranış Şekil 3.14'te gösterilmiştir. Sürüler halinde yaşayan atlardan bir kısmı lider, kalanı takipçi attır. Bu durumda, HOA'daki Hiyerarşi davranışı sürüdeki en iyi atın takip edilmesi şeklinde yorumlanır. Hiyerarşiye uyan atlar 5-15 yaş aralığındaki yetişkin atlardır. Çalışmalarda nitelendirilen orta yaş  $\beta$  ve  $\gamma$  atlarının (5-15 yaş arası) hiyerarşi yasasına uyduğu görülmüştür. Bu yüzden at sürüsünde en iyi atın yaşı ve konumu önemlidir. Tez çalışmasında önerilen algoritmada hiyerarşi davranışı oluşturulacak bireylerin en iyi konuma doğru hızlanmasını/yönlendirilmesini temsil etmektedir.



Şekil 3.14. Atların hiyerarşi davranışı.

### 3.6.1.3. Sosyallik

Atlar sosyal bir yaşam sürdürdüklerinden başka hayvanlarla birlikte yaşarlar. Bazı atlar gruplar halinde yaşarken bazıları da diğer hayvanlarla savaşırlar. Atlar, avcılar ve çeşitli hayvanlar tarafından avlanabilirler. Bu sebeple sürü halinde yaşamak atların güvenliği açısından önem taşımaktadır. Atların birlikte yaşadığı diğer hayvanlarla birlikte temasta olduğu davranışa sosyallik davranışı denir. Sosyal olarak kalabalık olarak yaşayan atların tehlikeler karşısında güvende ve hayatta kalma şansı artar. Bu davranış, atların yalnız kalmadığı ve diğer atların ortalama pozisyonuna doğru hareket ettiği bir davranış olarak değerlendirilir. Sosyallik davranışı 5-15 yaşlarındaki atlar tarafından yapılır. Tez çalışmasında önerilen algoritmada Sosyallik davranışı, keşif amacıyla olan bireylerin, diğer atların ortalama konumuna doğru bir hareket olarak kabul edilir ve mevcut durumun ortalamasını göz önüne alarak yönlendirilmesini sağlamaktadır. Bu davranış Şekil 3.15'te gösterilmiştir.



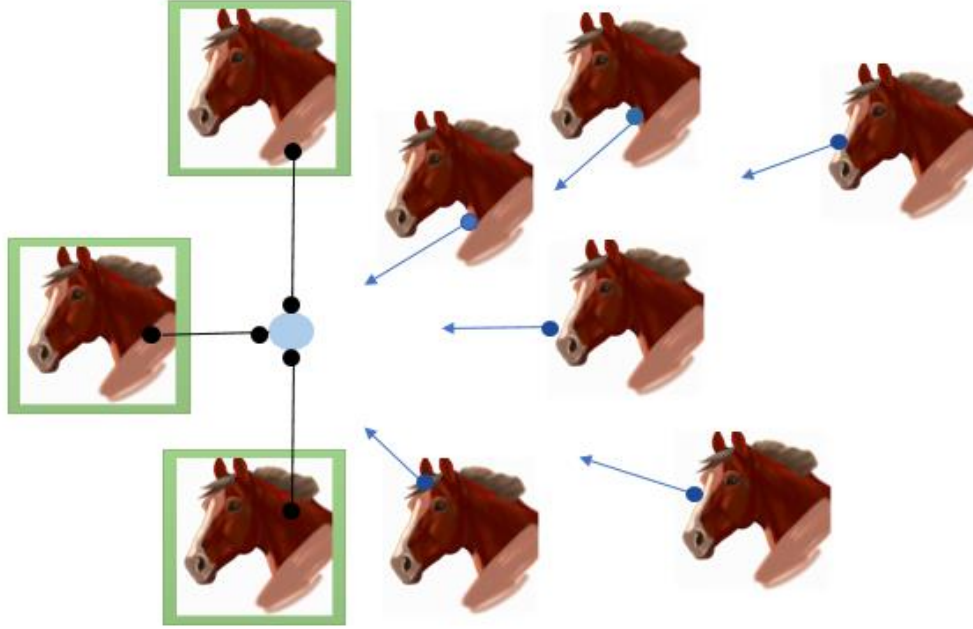
Şekil 3.15. Atların sosyalleşme davranışı.

### 3.6.1.4. Taklit

Atlar birbirini taklit eder. Genel olarak genç atlar, meraların doğru pozisyonunu ve uygun otlama yeri bulma, savunma mekanizması geliştirme vb. gibi güvenli bir ortamda kalmalarını sağlayan davranışları yetişkin atlardan öğrenirler. Tüm sürülerde olduğu gibi at sürüsünde de iyi veya kötü huylar taklit edilebilir. Genç atların diğer atları taklit ettiği bu süreç, atlar olgunlaşmıca kadar devam eder. Atların taklit



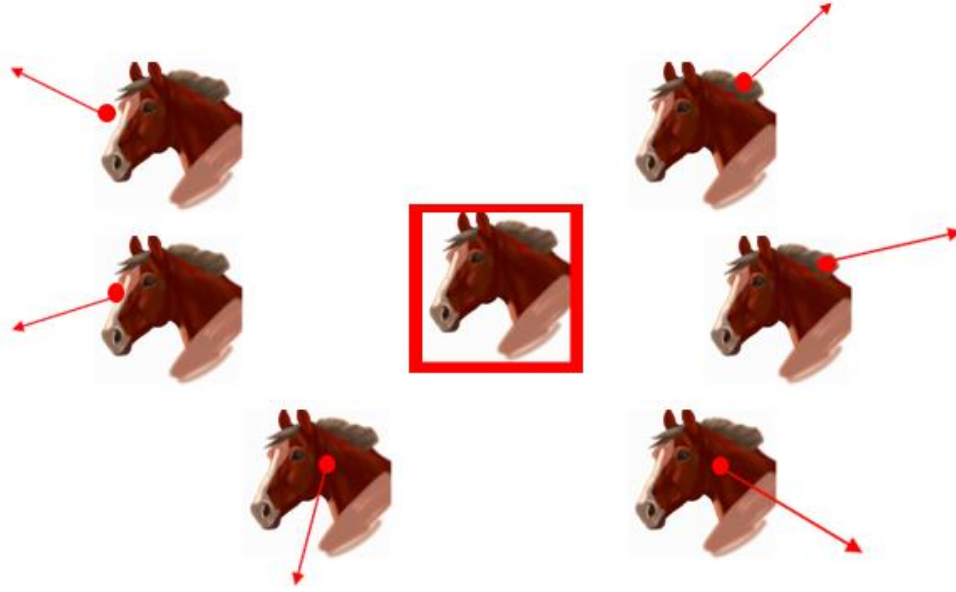
davranışı da önerilen algoritmada en iyi çözümü bulmak için kullanılmıştır. Bu davranış Şekil 3.16'da gösterilmiştir.



Şekil 3.16. Atların taklit etme davranışı.

### 3.6.1.5. Savunma

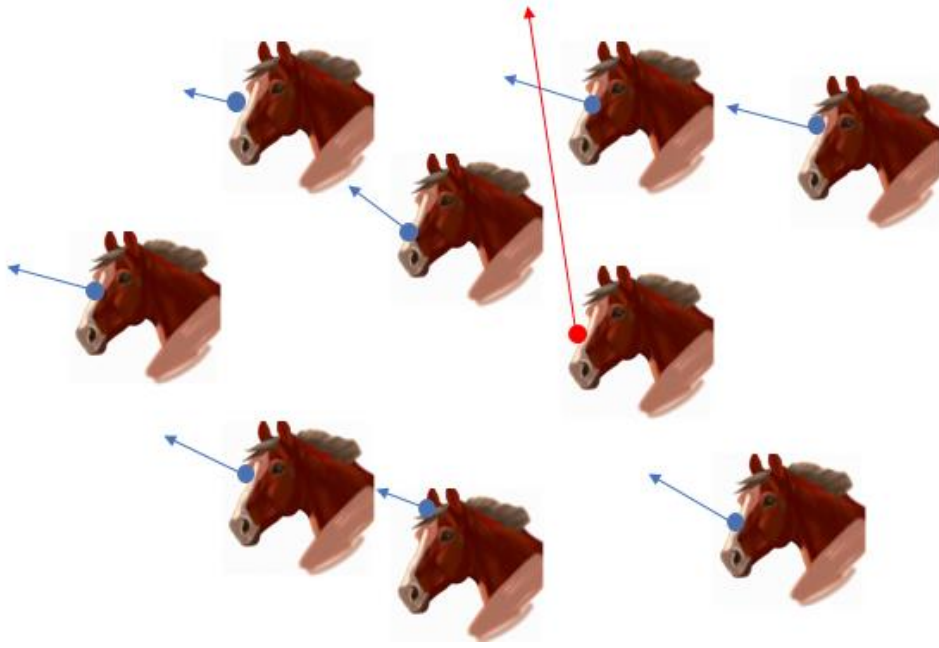
Atların savunma mekanizması iki şekildedir. Atlar ya saldırıdan kaçar ya da saldırganla uyum içinde yaşar. Ancak ilk tepkileri kaçmak olur. Ayrıca, tuzağa düşme durumunda geriye giderler. Atlar, içgüdüsel olarak kurtlar gibi düşmanların olduğu tehlikeli ortamlardan kaçar, rakiplerini uzaklaştırır ve yiyecek, su için savaşır. Böyle bir savunma mekanizması, mümkün olduğunda, genç veya olgun bir atın tüm yaşamı boyunca mevcuttur. Atların savunma mekanizması, önerilen algoritmada atı uygun olmayan pozisyonlardan uzak tutmak için kullanılmıştır. Bu davranış Şekil 3.17'de gösterilmiştir.



Şekil 3.17. Atların savunma davranışı.

### 3.6.1.6. Dolaşım

Atlar yaratılış itibari ile meraklı hayvanlardır. Yeni yerler keşfetmek, yaşam alanlarını tanımak amaçlı gidebildikleri her yeri ziyaret ederler. Atlar yiyecek bulmak için meradan meraya doğada dolaşırlar ve otlanırlar. Bir at otlamak için aniden başka bir yere gidebilir. Bu davranış önerilen algorithmada rastgele bir hareket olarak karşılık bulmuştur. Atlarda gezinme hemen hemen her yaşta yaşlarda görülür ve olgunluğa eriştikçe yavaş yavaş kaybolur. Önerilen algorithmada dolaşma davranışı yerel bir arama yapma durumu ve yerel minimumdan kaçış durumu simgelerken, atın dolaşması rastgele hız vektörünü temsil eder. Bu davranış Şekil 3.18’de gösterilmiştir.



Şekil 3.18. Atların dolaşma davranışı.

Yukarıdaki bahsedilen davranış kalıpları ile atların farklı yaşlardaki hız vektörünü hesaplamak için aşağıdaki matematiksel denklemler kullanılır.

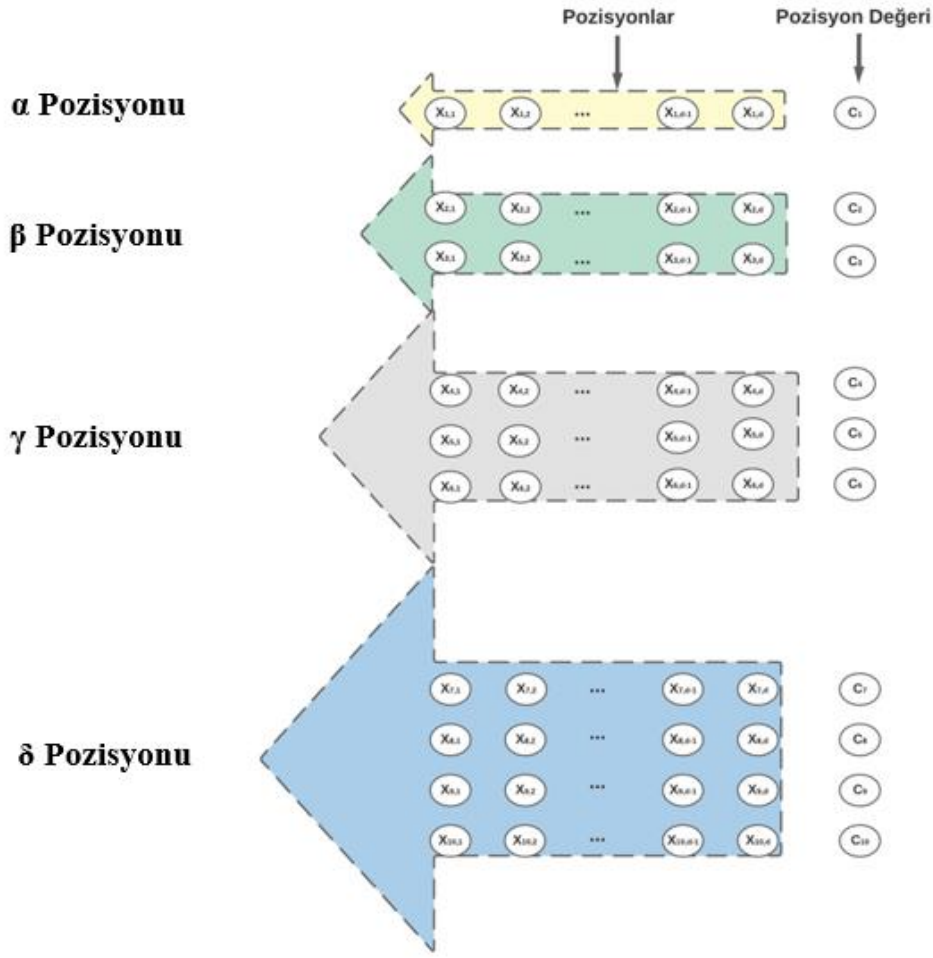
$$\vec{V}_m^{t,\alpha} = \vec{G}_m^{t,\alpha} + \vec{D}_m^{t,\alpha} \quad (3.10)$$

$$\vec{V}_m^{t,\beta} = \vec{G}_m^{t,\beta} + \vec{D}_m^{t,\beta} + \vec{H}_m^{t,\beta} + \vec{S}_m^{t,\beta} \quad (3.11)$$

$$\vec{V}_m^{t,\gamma} = \vec{G}_m^{t,\gamma} + \vec{D}_m^{t,\gamma} + \vec{H}_m^{t,\gamma} + \vec{S}_m^{t,\gamma} + \vec{I}_m^{t,\gamma} + \vec{R}_m^{t,\gamma} \quad (3.12)$$

$$\vec{V}_m^{t,\delta} = \vec{G}_m^{t,\delta} + \vec{I}_m^{t,\delta} + \vec{R}_m^{t,\delta} \quad (3.13)$$

Popülasyondaki atların ilk %10'u  $\alpha$  atları olarak seçilir, sonraki ilk %20'lik dilimin yaşı  $\beta$  atlar olarak atanır, atların bir sonraki %30'u  $\gamma$  atları olarak belirtilir ve son %40 atların yaşı  $\delta$  olarak atanır. Popülasyonlardaki atlar davranışlarına göre kategorileştirilirken uygunluk değeri açısından en iyiden en kötüye doğru sıralanmaktadır. Önerilen algoritmada atların pozisyon maliyetlerine göre etiketlenme/sıralanma süreçleri Şekil 3.19'da gösterilmiştir.



Şekil 3.19. HOA algoritmasının sıralama süreci.

### 3.6.2. Öznitelik seçme problemi için ikili at sürüsü algoritması

Metasezgisel yöntemler öznitelik seçim probleminde kullanılan yaklaşımlardan biridir. Literatürde, birçok öznitelik seçme yöntemi, optimal altküme için metasezgisel algoritmaları benimsemiştir (Mafarja vd., 2017a). Orjinal HOA sürekli uzaydaki optimizasyon problemlerini çözen ve büyük boyutlu veride de başarılı olan bir metasezgisel algoritmadır (MiarNaeimi vd., 2021b). Tezin bu kısmında öznitelik seçim problemine uygulanmak üzere tasarlanan ve uygun öznitelik alt kümesini seçmesi için geliştirilen İkili At Sürüsü Algoritması açıklanacaktır. HOA ile aynı stratejiler İkili HOA içinde uygulanır.

Önerilen yeni yaklaşımda öznitelik seçimi 0 veya 1 değerlerinde oluşan ikili bir vektör olarak temsil edilir. Atların konumu N boyutlu bir veri seti için N uzunluğunda ikili

bir bit dizisi ile temsil edilir. Her bit bir özelliğin seçilip seçilmediğini temsil ederken, bite ait değer "1" olması o özelliğin seçildiği, "0" olması ise o özelliğin seçilmediğini ifade eder. Oluşturulan her konum ilgili veri setinden oluşturulan bir özellik alt kümesidir. Algoritma performansını iyileştirmek ve en iyi çözümü elde edebilmek için keşif ve sömürü arasında denge kurulması gerekmektedir (Qasim vd., 2020).

HOA'nın ikili uzayda çalışabilmesi için atların konum ve pozisyon değerlerinin sürekli değerler yerine ikili değerler formatında olması gerekmektedir. Sürekli uzaydaki değerler çeşitli transfer fonksiyonları kullanılarak ikili uzaya dönüştürülebilir. Öznitelik seçimi için önerilen İkili At Sürüsü Algoritması (BHOAFS) Şekil 3.12'de verilen temel HOA'nın tüm adımlarını kullanmaktadır. Buna ek olarak, sürekli değerleri ikili değerlere dönüştürmek için U-şekilli transfer fonksiyonu (Mirjalili vd., 2020) kullanılmıştır. Transfer fonksiyonu ile ayrıklaştırma sürecinde; BHOAFS algoritması öncelikle temel HOA algoritmasındaki denklem 3.9'u kullanmaktadır. Daha sonra ise denklem 3.14'ü kullanarak sürekli değerleri ayrıklaştırmaktadır.

$$U(X_{m,j}^t) = \delta |X_{m,j}^t|^\alpha \quad (3.14)$$

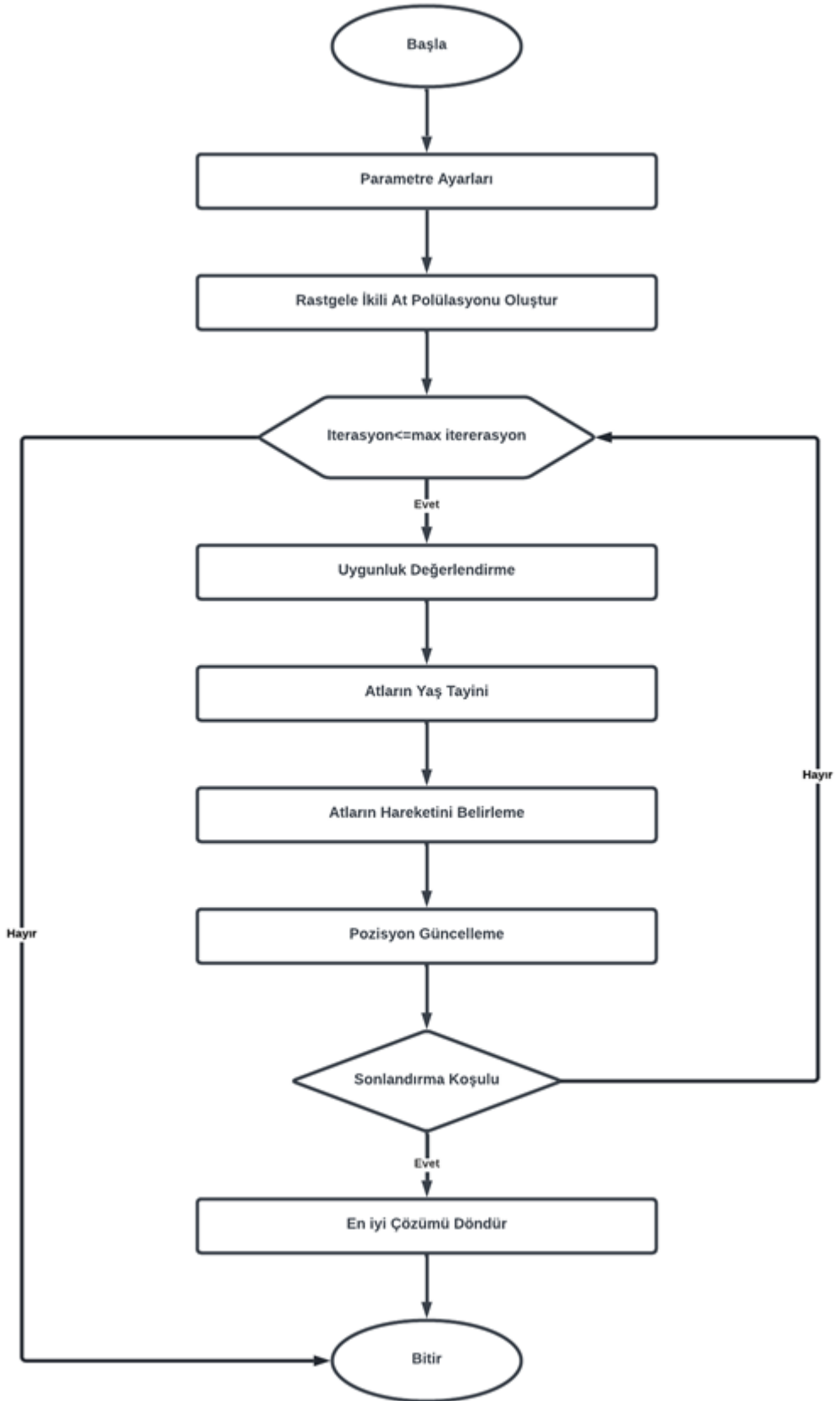
Denklem 3.14'teki U-şekilli transfer fonksiyonu,  $\delta$  ve  $\alpha$  olmak üzere iki kontrol parametresi ile tasarlanmıştır;  $\delta$  transfer fonksiyonunun eğimini ve  $\alpha$  U-şekilli transfer fonksiyonunun havzasının genişliğini tanımlar.

U-şekilli transfer fonksiyonu da  $X_{m,j}$ 'nin mevcut değerini '0' ya da '1' olacak şekilde ikili bir ayrık değere dönüştürmektedir.  $X_{k,j}$ 'nin '0' ya da '1' değerine dönüştürme işlemi denklem 3.15'te verilmiştir. Son olarak aday çözümün ilgili boyutu için yeni konumu belirlemektedir.

$$X_{k,j}^t = \begin{cases} 1 & rand > U(X_{m,j}^t), \\ 0 & rand < U(X_{m,j}^t). \end{cases} \quad (3.15)$$

BHOAFS Algoritması için akış şeması Şekil 3.20'de görülmektedir. Algoritma seçilen veri setleri için yapılacak olan ön işleme adımı ile başlar. İlgili veriler uygun hale getirildikten sonra algoritma nitelik seçimine başlar. Belirlenen iterasyon boyunca seçilen sınıflandırma algoritması ve uygunluk fonksiyonu sayesinde algoritma çevrimi devam eder. Şekil 3.20'deki gibi algoritmanın durma kriteri sağlandığında, en optimal

özelliklerin seçildiği kabul edilerek bu özelliklerin indirgenmiş hali test veri setine uygulanır. Test veri setinin boyutu nitelik seçim işlemi sonucu eğitim veri setinin ürettiği alt küme boyutu ile azaltılır. İndirgenmiş test seti de kullanılan sınıflandırma algoritması ile sınıflandırılarak, sonuçlar karşılaştırılır. Uygulama sonrası niteliklerin en az sayıda, test veri seti sınıflandırma doğruluğunun yüksek ve hata oranının düşük olması beklenir. Bu beklenti karşılandığında algoritma sonlandırılarak, iyi bilinen diğer metasezgisel nitelik seçimi algoritmalar ile kıyaslanabilir.



Şekil 3.20. Önerilen BHOAFS için akış şeması.

### 3.6.2.1. İkili arama uzayında çözüm popülasyonun tasarımı

At Sürüsü Algoritması başlangıçta rastgele oluşturulan atlardan bir popülasyon yaratarak matematiksel olarak tanımlanır. Oluşturulan bu araştırma alanında her  $X_d = (X_{d1}, X_{d2}, X_{d3}, \dots, X_{dn})$  olmak üzere  $n$ . pozisyonundaki atı temsil eder.

$$d = 1, 2, \dots, N$$

$$j = 1, 2, \dots, D$$

$U(0, 1)$ , 0 ile 1 arasında düzgün dağılmış rastgele bir değer üreten fonksiyon,

$X_j^{min}$  ve  $X_j^{max}$   $j$  konumundaki arama uzayının sınırları olmak üzere olmak üzere ;

$$X_{d,j} = X_j^{min} + (X_j^{max} - X_j^{min}) * U(0,1) \quad (3.16)$$

denklem 3.16'da ifade edilir.  $X_1, X_2, X_3, X_4, \dots, X_n$  her bir  $X$  atın konumunu temsil eden bir vektör dizisini,  $x_i^j$   $X_j$  atının  $i$ . bitini göstermek üzere; arama uzayındaki at popülasyonun ikili temsili denklem 3.17'deki matris ile gösterilir.

$$\text{At Popülasyonu} = \begin{bmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,D} \\ X_{2,1} & X_{2,2} & \dots & X_{2,D} \\ \vdots & \vdots & \dots & \vdots \\ X_{N,1} & X_{N,2} & \dots & X_{N,D} \end{bmatrix} \quad (3.17)$$

$X_i$  çözümü  $X_i = [0.59, 0.17, 0.32, 0.95, 0.85, 0.50, 0.77]$  olmak üzere

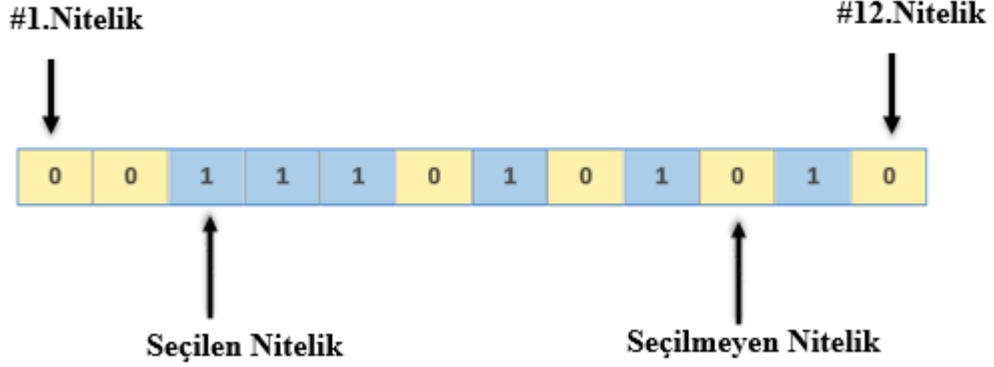
İkili  $X_i = [1, 0, 0, 1, 1, 0, 1]$  olarak temsil edilir. İkili  $X_i$  'ye ait bu temsili daha net açıklarsak, ikili diziden birinci, dördüncü, beşinci ve yedinci özelliklerin oluşturulan özellik alt kümesinden seçilmesi gerektiği ikinci, üçüncü ve altıncı özelliklerin seçilmemesi (kullanılmaması) gerektiğini ifade eder. Yani ikinci, üçüncü ve altıncı özellikler ortadan kaldırılmalıdır. Bu durum Şekil 3.21 ile örneklenmiştir.

İkili olarak temsil edilen arama uzayında BHOAFS'deki atların hareketi aşağıdaki denklem 3.18 ile hesaplanır.

$$x_i^{k+1} = x_i^k + \theta * (x_{BH}^k - x_i^k) \quad i = 1, 2, \dots, N \quad (3.18)$$



burada  $x_i^k$  ve  $x_i^{k+1}$ , i'nci atın sırasıyla k ve k+1 iterasyonundaki konumlarıdır.  $x_{BH}^k$ , en iyi atın arama uzaydaki konumudur.  $\theta$ , 0 ile 1 arasında düzgün dağılmış rastgele bir sayı ve N ise atların sayısıdır (Awadallah vd., 2022).



Şekil 3.21. Temsili bir ikili çözüm vektörü (Turabieh, 2019).

### 3.6.2.2. Sınıflandırıcı algoritması ve uygunluk değerlendirme

Uygun öznelik alt kümesi elde edilmesi için oluşturulan çözümlerin değerlendirilmesi gerekmektedir. Öznelik belirlerken ikili operatörlerle çalışan algoritmalar çözüm uzayında 0 ve 1'ler ile çalıştığından sürekli uzayda çalışan algoritmalarından daha başarılıdır. Her iterasyonda, algoritma seçilen öznelikleri kullanarak oluşturulan alt kümenin başarısı k-NN, SVM gibi bir sınıflandırıcı ile değerlendirilir. Önerilen algoritmada k-NN ve SVM gibi bir sınıflandırıcıları kullanılmıştır.

**k-NN Sınıflandırıcı:** k-NN sınıflandırması için veriler k=5 ile sınıflandırılmış ve eğitim seti sonuçları 10 kat çapraz doğrulama ile elde edilmiştir.

**SVM Sınıflandırıcı:** SVM için RBF Kernel yöntemi kullanılmıştır. Kernel Trick yöntemi ile RBF Kernel adı verilen yöntem, normal dağılımla her bir noktanın belirli bir noktaya ne kadar benzer olduğunu hesaplar. Çok sınıflı veri kümeleri için, SVM bire karşı hepsi stratejisiyle uygulanmıştır. Eğitim seti sonuçları 10 kat çapraz doğrulama ile elde edilmiştir.

Sınıflandırıcılar kullanılarak yapılan değerlendirme veri setleri için çalıştırılarak, hangi sınıflandırma algoritmasının daha başarılı olduğu tespit edilmiştir. Atların konum vektöründeki sürekli değerler ikili değerlere dönüştürülürken, geliştirilen

algoritmada uygunluk fonksiyonu, hesaplanan değer ve kalitesine göre at popülasyonundaki ikili tanımlanan at çözümlerindeki bitsel işlemin 0'dan 1'e ya da 1'den 0'a dönüştürülmesi gerektiğine karar verilmektedir (Kennedy & Eberhart, 1997). Literatüre göre sürekli arama uzayından ikili arama uzayına dönüştürülen algoritmalar için aday bir çözümün uygunluğu hesaplanırken, alt kümedeki öznitelik sayısı da göz önüne alınmalıdır. Ancak test kümesindeki hatayı artırmadan öznitelik sayısını azaltmak hedeflendiğinden, hem doğruluğun hem de ilgili veri setindeki özniteliklerin sayısı ile indirgenmiş veri setinin öznitelik sayısı kullanılarak Uygunluk Fonksiyonu oluşturulmalıdır (Mafarja vd., 2017b), (Baş & Ülker, 2020).

İkili optimizasyon yönteminin amacı değerlendirilen bu sınıflandırma başarısını arttırırken, seçilen öznitelik sayısını en aza indirmektedir. Kısaca başarı en yüksek sınıflandırma doğruluğu elde ederken, en az öznitelik seçimi olarak değerlendirilir. İki ayrı amacın başarılması gereken bu durum çok amaçlı bir problem olarak değerlendirilir. Uygunluk fonksiyon formülü ise en yüksek sınıflandırma doğruluğuna erişmek ve en az öznitelik seçimini oluşturacak şekilde denklem 3.19'da ifade edildiği gibi uygulanmıştır.

$S = [0,1]$  arasında sabit bir değer

$B = 1-S,$

$D =$  ilgili sınıflandırma algoritmasına ait doğruluk değeri,

$R =$  seçilen öznitelik sayısı,

$C =$  toplam öznitelik sayısı olmak üzere

algoritma için uygulanacak olan uygunluk değeri denklem 3.19'daki gibi ifade edilir.

$$\text{Uygunluk değeri } f(X_m) = S * D + B * \frac{C - R}{C} \quad (3.19)$$

Atların BHOAFS'deki konumları oluşturulduktan sonra, bu konumlara ait uygunluk değeri  $f(X_m)$ ,  $\forall m = 1, 2, \dots, N$  olmak üzere uygunluk fonksiyonu kullanılarak hesaplanır. Uygunluk fonksiyonun en yüksek değerini elde edecek şekilde oluşturulan öznitelik alt kümesi hedeflenen ideal alt küme olarak kabul edilecektir. At sürüsündeki her at popülasyona ait yeni bir çözüm ürettiğinde bu çözümün değerlendirmesi sınıflandırıcılar ve uygunluk fonksiyonu ile tekrar hesaplanmıştır.

### 3.6.2.3. Atların arama stratejisi

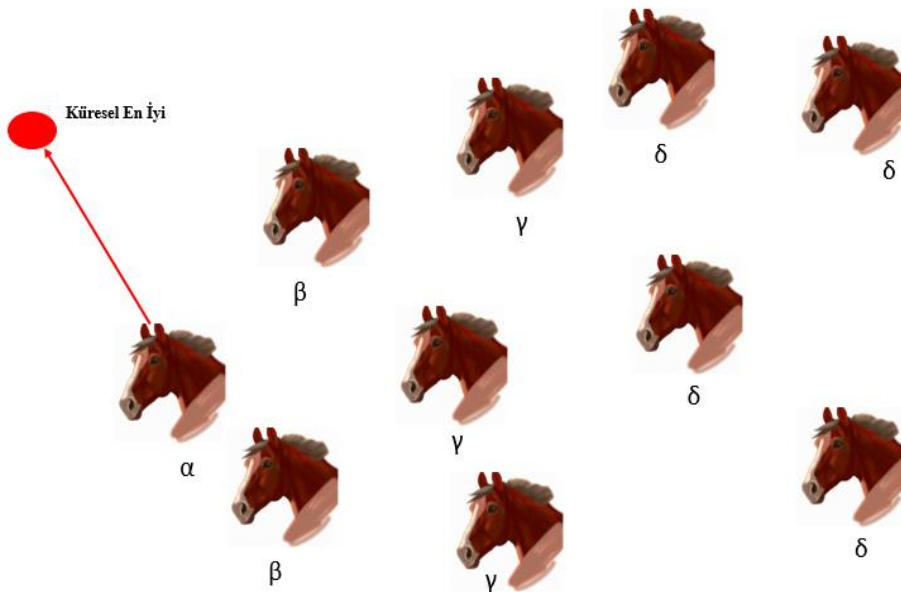
Atların yaşam döngüsü içindeki davranışları yaşları ile bağıntılı olarak çeşitli şekillerde sembolize edilmiştir. HOA algoritmasındaki davranış durumları BHOAFS algoritması ile aşağıdaki gibi ilişkilendirilmiştir.

- $\beta$  ve  $\gamma$  atları için hiyerarşi faktörü
- $\beta$  ve  $\gamma$  atları için sosyallik faktörü; atların iyi atların ortalama pozisyonuna konumlanması
- $\gamma$  atlar için taklit faktörü; en iyi atın pozisyonuna doğru konumlanması
- $\alpha$ ,  $\beta$  ve  $\gamma$  atları için savunma faktörü; atların kötü pozisyondan uzaklaşması
- $\delta$  ve  $\gamma$  atları için dolaşım faktörü; genel arama için atların farklı konumları dolaşması olarak temsil edilmiştir.

Popülasyonlardaki her at, yaşına göre aşağıdaki dört sınıftan birine ayrılır:  $\delta$ , 0-5 yaş arası atları,  $\gamma$  5 ila 10 yaş arası atları,  $\beta$  10 ila 15 yaş arası atları,  $\alpha$  ise 15 yaşından büyük atları temsil eder. Her yinelemede, aşağıdaki kurallara göre popülasyondaki atların yaşları güncellenir.

#### $\alpha$ atları

$\alpha$  atları sürünün o andaki en iyi konumunu temsil eder ve sürüyü yönlendirirler. Yerel arama stratejisindeki en önemli atlardır. Arama uzayındaki muhtemel en iyi çözüme yakın oldukları düşünüldüğünden  $\alpha$  atları, buldukları yerde ultra yüksek hassasiyetle küresel optimumu bulmak amaçlı yerel bir aramaya başlar. Küresel en iyi ile

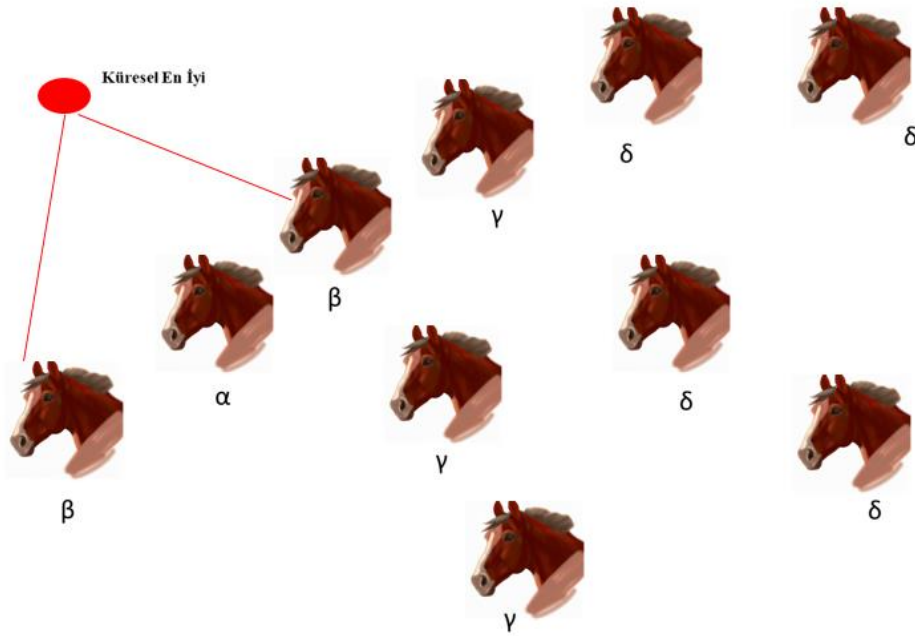


Şekil 3.22.  $\alpha$  atları için arama stratejisi

arasındaki mesafe ölçümü neticesinde bit değişikliği yapılarak bir sonraki iterasyona hazırlanıp yeniden konumlandırılır. Bu atlara ait temsili gösterim Şekil 3.22'de gösterilmiştir.

### **$\beta$ atları**

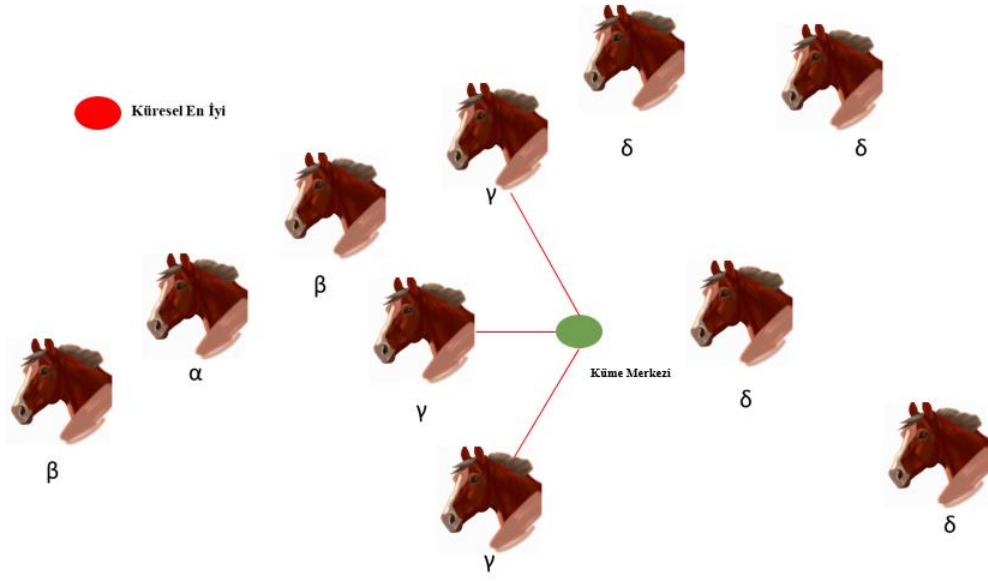
$\beta$  atları,  $\alpha$  atlarına en çok benzeyen atlardır ancak onlardan daha iyi pozisyonda olan optimal konumları dikkatlice aramakla görevlendirilmiştir.  $\beta$  atları  $\alpha$  atlarının komşuluğunu temsil etmektedir.  $\alpha$  atlarının etrafındaki optimumu bulmayı sağlar. Bu sebeple küresel en iyi ile her bir atın arasındaki mesafe Minimum Mesafe Fonksiyonu ile ölçülerek, belirlenen bir  $k$  katsayısı ile,  $k$  kadar bit değiştirilerek bir sonraki iterasyona aktarılırlar. Bu atlara ait temsili gösterim Şekil 3.23'te gösterilmiştir.



Şekil 3.23.  $\beta$  atları için arama stratejisi.

### **$\gamma$ atları**

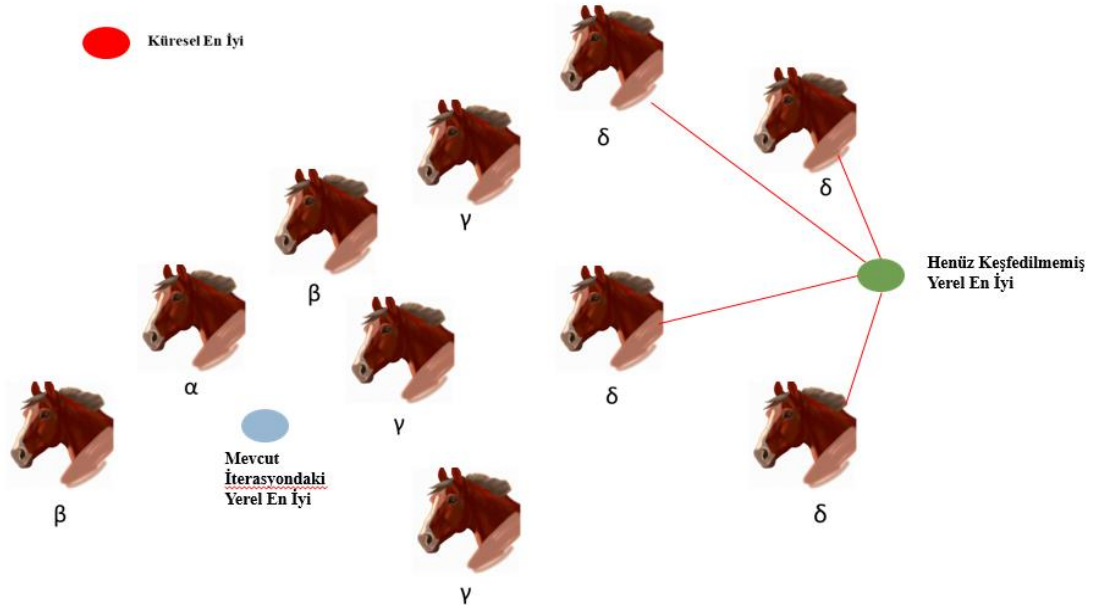
$\gamma$  atları  $\alpha$  atlarına doğru ilerlemezler. Bu sebeple yeni yerler arama ve yeni olası küresel optimum konumları bulma konusuyla daha ilgililerdir.  $\gamma$  atları küresel aramaya daha yatkın oldukları için mevcut  $\gamma$  atlarının Sürü Merkezi Bulma Fonksiyonu ile merkez noktası bulunarak, her bir  $\gamma$  atı ile arasındaki mesafe ölçümü yapılır. Bu ölçüye göre belirlenen  $k$  kadar bit değişikliği sayesinde küresel en iyiyi ararken sürünün çeşitlenmesini sağlarlar. Bu atlara ait temsili gösterim Şekil 3.24'te gösterilmiştir.



Şekil 3.24.  $\gamma$  atları için arama stratejisi.

### $\delta$ atları

$\delta$  atları pozisyon itibari ile sürünün o anki en kötü konumuna sahiptir. Bu konumu mevcut yerel optimuma doğru iyileştirmek yerine, başka yerel optimumları bulmak ve küresel optimuma ulaşabilmek için geliştirmek sürüdeki çeşitliliği artırır. Bu sebeple  $\delta$  atları keşif aşaması için daha uygundur.  $\delta$  atlarının her bir iterasyonda rastgele oluşturulan bir değerle bitisel işleme girerek sürünün keşif kabiliyetinin artırılması amaçlanmaktadır. Bu atlara ait temsili gösterim Şekil 3.25'te gösterilmiştir.



Şekil 3.25.  $\delta$  atları için arama stratejisi.

Atlara ait genel arama stratejilerinin özet hali Tablo 3.2’de gösterilmiştir.

**Tablo 3.2.** Atların arama stratejileri genel bakış.

Atların Davranış Modeli	Otlama	Hiyerarşi	Sosyallik	Taklit	Savunma	Dolaşma	Önerilen Algoritma İşlemi	Etkili Arama Yöntemi
$\alpha$ atları	X				X		En iyi Pozisyonda bulunan at olduğu için yerel aramayla küresel en iyiden farklı bir bitini değiştirerek konumlanır.	Yerel Arama
$\beta$ atları	X	X	X		X		Küresel en iyi ile her $\beta$ atının arasındaki mesafe ölçülerek k kadar biti değiştirilerek konumlanır.	Yerel ve Küresel Arama
$\gamma$ atları	X	X	X	X	X	X	$\gamma$ atlarının merkez noktası bulunarak, her bir $\gamma$ atı ile arasındaki mesafe ölçümü yapılır. Bu ölçüme göre sürünün alakasız bir bölgeyi değil mevcut bölgeyi iyi tarayacak şekilde k kadar biti değiştirilerek konumlanır.	Yerel ve Küresel Arama

**Tablo 3.2.(Devamı)** Atların arama stratejileri genel bakış.

Atların Davranış Modeli	Otlama	Hiyerarşi	Sosyallik	Taklit	Savunma	Dolaşma	Önerilen Algoritma İşlemi	Etkili Arama Yöntemi
δ atları	X			X		X	δ atları her bir iterasyonda rastgele oluşturulan bir değerle bitsel işleme girerek sürünün keşif kabiliyetinin artırır.	Küresel Arama



#### 3.6.2.4. Yerel arama tabanlı pozisyon güncelleme stratejisi

Her at bulunduğu konum itibari ile iki amaca hizmet eder. Bu amaçlar:

- Bulunduğu konumuna ait yerel arama yapmak ve bu konumun en iyisini çözüm olarak işaretlemek,
- İterasyon boyunca bulunan çözümlerin en iyisini seçmektir.

Atlar iterasyona rastgele konumlanarak başlarlar. Her at, Tablo 3.2'teki davranışlardan hangisi ile etiketli ise o davranışa ait işlemleri gerçekleştirir. Bu davranışları değerlendirmek ve bir sonraki pozisyonu belirleyebilmek için uygunluk fonksiyonu kullanılır.

Tasarlanan BHOAFS Algoritmasında her bir at öznitelik seçimi probleminin olası bir çözümü, at pozisyonuna göre elde edilen öznitelik alt kümesi ile çalıştırılan sınıflandırma algoritmasının doğruluğu ise çözümün kalitesini belirtir. Atlar pozisyonuna göre yerel çözümler sunarlar, her bir çözüm de atın kalitesi ile ilişkilendirilir. Yerel çözümlerin en iyileri iterasyon boyunca güncellenir.  $\alpha$  ve  $\beta$  atları keşfedilen çözümlerden faydalanma (exploitation) işleminde,  $\delta$  ve  $\gamma$  atları ise keşif (exploration) sürecinde görev alır.

#### 3.6.2.5. Sürü merkezi bulma fonksiyonu(SMF)

Önerilen algoritma için hem genel arama stratejisi ile henüz keşfedilmemiş noktaların keşfi hem de çözüm uzayında dolaşırken mevcut durumdan alakasız noktalara gitmeyi engellemek amacıyla tasarlanan fonksiyonda, sürü içindeki bazı bireylerin merkez noktası bulunarak, seçilen bireyler ile aralarındaki mesafe ölçümü yapılır. Bireyler belirlenen bir  $k$  sabiti ile mevcut konumlarından belirli uzaklıktaki başka bir noktaya yönlendirilir. Bu işlem sayesinde küresel en iyiyi ararken sürünün çeşitlenmesi de sağlanır.

Eşit uzunluktaki iki dizi arasındaki Hamming mesafesi, birbirlerine karşılık gelen sembollerin farklı olduğu konumların sayısıdır. Bu nedenle Hamming mesafesi iki at arasındaki mesafeyi ölçmek için kullanılabilir.  $X$  ve  $Y$  iki atın konumu temsil eden ikili bit dizeleri olmak üzere;

$X$  ve  $Y$  atlarının birbirleri ile olan mesafesi denklem 3.20'de görüldüğü üzere  $X \text{ XOR } Y$  işlemi ile ortaya çıkan sonuca eşittir (Y. Chen vd., 2015).

$$h(X, Y) = \sum_{i=1}^N (x_i \oplus y_i) \quad (3.20)$$

$X^1, X^2, \dots, X^n$  her bir  $X$  atın konumunu temsil eden bir vektör dizisini,  $x_i^j$   $X^j$  atının  $i$ . bitini göstermek üzere;  $X_c$   $n$  adet atın küme merkezini gösterir ve denklem 3.21 ile temsil edilir (Kohavi & John, 1997).

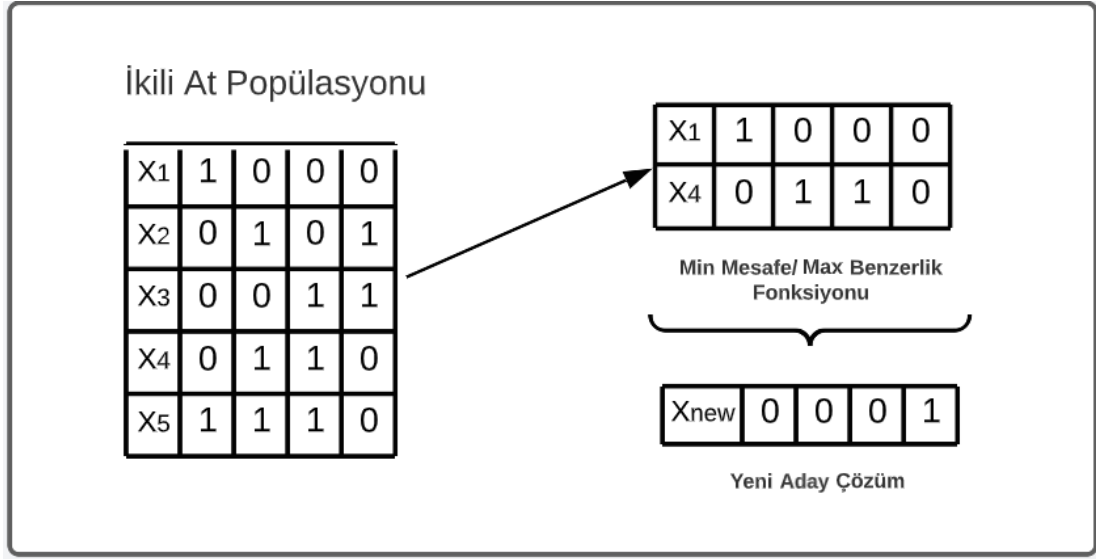
$$X_c = \{c_1, c_2, \dots, c_N \mid \text{Eğer } \frac{1}{n} \sum_{j=1}^N (x_i^j > 0.5) \text{ } c_i = 1 \text{ } \\ \text{değilse } c_i = 0 \} \quad (3.21)$$

Atlar iterasyona rastgele konumlanarak başlarlar. Her at, pozisyonuna uygun davranışa ait işlemleri gerçekleştirir. Bu davranışları değerlendirmek ve bir sonraki pozisyonu belirleyebilmek için uygunluk fonksiyonu kullanılır.

### 3.6.2.6. Minimum mesafe fonksiyonu

Önerilen algoritma için yerel arama stratejisi amacıyla tasarlanarak iki aday atın birbirine olan benzerliğini/yakınlığı ölçen fonksiyonda, sürü içindeki konumları iyi olan bireylerin mevcut konumları kaybedilmeden iyileştirme yapılarak bir sonraki iterasyona iletilme amacı güdülmüştür. Birbirine en yakın veya komşu olan iki kaynağın mesafesi belirlenen bir  $k$  değerinden küçükse, bu kaynaklardan uygunluk değeri daha düşük olan yeniden oluşturulur. Fonksiyonun çalışma mantığı Şekil 3.26'da temsil edilmiştir. Bu işlem her iterasyonda seçilen bireyler için tekrar yapılır. Buradaki amaç sürünün ilklendirilmesi sırasında, rastgele değil o ana kadarki gelinen optimal nokta potansiyelini kaybetmeden mevcut kaynağı yeniden oluşturmaktır. Böylelikle sürü ile ilgili en önemli sorun olan erken yakınsama önlenir ve çeşitlilik artırılır. Ayrıca, önerilen fonksiyon ile küresel arama kapasitesi azaltılmadan daha kapsamlı yerel arama yapılması sağlanmıştır.

Literatüre göre  $k$  değeri öznitelik seçimi yapılacak veri seti boyutu  $N$  olmak üzere  $N/3$  veya  $N/2$  olarak belirlenmiştir. Önerilen algoritmada veri setinin yüksek boyutlu olduğu durumlar için  $N/3$  olarak kullanılmıştır (ZAIMOGLU vd., 2020).



Şekil 3.26. Minimum mesafe fonksiyonu.

### 3.6.2.7. Kaotik ikili HOA algoritması

Kaotik değişkenler başlangıç koşullarına göre belirli aralıklardaki tüm durumları tekrar etmeden geçebilirler. Kaotik sayı dizilerinin yapılandırılması ve oluşturulması hızlı ve kolaydır (Abdel-Raouf vd., 2014). Bu davranışlarından dolayı kaos araması, rastgele aramaya göre tepeye tırmanma ve yerel optimumdan kaçma konusunda daha iyidir. Bu nedenle bir çok optimizasyon algoritmasında temel hareket işlevini veya denklemlerde kullanılan stokastik değişkenleri güncellemek için kaos haritaları kullanılmıştır. Literatür çalışmaları incelendiğinde, kaotik haritalar optimizasyon algoritmalarında iki şekilde uygulanmıştır :

Rastgele Sayı Üreticisi: Başlangıç popülasyonunun oluşturulmasında, sezgisel denklemlerde katsayı olarak ya da birey seçimi yaparken olasılık değeri olarak kaotik sistemlerden üretilen değerler kullanılabilir.

Yerel Arama Tekniği : Kaotik arama, arama davranışını zenginleştirmek, küresel yakınsamanın iyileştirmek ve yerel optimuma sıkışıp kalmamak için metasezgisel prosedürlerin içine dahil edilmektedir (Ewees vd., 2019).

Öznitelik seçim problemi için önerilen Kaotik Haritalı İkili HOA Algoritmasında (BCHOAFS) BHOAFS algoritmasının nitelik seçim performansını iyileştirmek ve kararlılığını arttırmak amacıyla tek boyutlu beş kaotik harita rastgele sayı üretici olarak arama işlemi başlangıcında kullanılmıştır. Önerilen BCHOAFS algoritması için

seçilen kaotik haritaların tümü için başlangıç değeri 0.7 olarak ayarlanmıştır (Sayed, Hassanien, vd., 2019a). BCHOAFS algoritması, kaotik haritalar kullanılarak N adet at içeren rastgele bir sayı dizisi üretir. Her at, verilen öznitelik seçim problemi için popülasyonda bir çözümü temsil eder. Popülasyon ise denklem 3.22 kullanılarak oluşturulur:

$$x_i^{k+1} = x_i^k + C_{map} * (x_{BestHorse}^k - x_i^k) \quad i = 1, 2, \dots, N \quad (3.22)$$

burada  $x_i^k$  ve  $x_i^{k+1}$ , i'nci atın sırasıyla k ve k+1 iterasyonundaki konumlarıdır.  $x_{BestHorse}^k$ , e iyi atın uzaydaki konumudur, ve N ise atların sayısıdır.  $C_{map}$  ise kaotik harita ile üretilmiş  $\in [0, 1]$  değeridir. Her arama uzayında keşif ve sömürü aşamaları için iyi sonuçlar elde etmek ve optimal küresel çözümü bulmak amaçlı önerilen algoritmanın performansını iyileştirmek için  $\theta$  parametresi, kaotik haritadan elde edilen değerle  $C_{map}$  değeri ile denklem 3.22'teki gibi değiştirilir. Şekil 3.27'de BCHOAFS sözde kodu görülmektedir.

```

1: Başla
2:  $C_{map}$  olarak kaotik haritalar oluştur ve popülasyonları aday at konumlarıyla başlat
    $X_i, i = 1, 2, \dots, n$ 
3: n atlık bir at popülasyonu oluştur ( $X_0$ )
4: Her at çözümünün uygunluk değerini hesapla (Denklemler 3.19)
5: En iyi at çözümü olarak en iyi uygunluk değerine sahip olan atı seç ( $X_{best}$ )
6: while (iter <= Maxiter) do
7:     for each popülasyondaki her at için do
8:         Her X atını uygunluk fonksiyonu = FitnessXi ile değerlendir
           ve FitnessXi'yi bul (Denklemler 3.19)
9:         if Fitness ( $X_i$ ) >= Fitness( $X_{best}$ ) ve  $|X_i| < |X_{best}|$  ise
10:             $X_{best} = X_i$  ve Fitness( $X_{best}$ ) = Fitness ( $X_i$ )
11:            SMF (Denklemler 3.21)
12:        end if
13:        if SMF ( $X_{best} - X_i$ ) < SMF ( $X_{best}$ )
14:            Çözüm uzayında X'i yeni oluşturulan  $X_{best}$  ile değiştir
15:        end if
16:    end for
16:    for j = 1: numbers of horse
18:        for k = 1: numbers of horse
19:            Calculate  $X_i^{k+1} = X_i^k + C_{map} * (X_{best} - X_i^k), i = 1, \dots, n$ 
20:            if  $C_{map} < 0.5$ 
21:                Set  $X^{new} = 0$ 
22:            else
23:                Set  $X^{new} = 0$ 
24:            end if
25:        end k;
26:    end j;
27:    Set iter = iter +1
28:end while
29:return best solution
30:end

```

Şekil 3.27. BCHOAFS sözde kodu.

### 3.6.2.8. BHOAFS algoritması temel kabulleri

- Tasarlanan BHOAFS Algoritmasında her bir at nitelik seçimi probleminin olası bir çözümü, at pozisyonuna göre elde edilen öznitelik alt kümesi ile çalıştırılan sınıflandırma algoritmasının doğruluğu ise çözümün kalitesini belirtir.
- Atlar pozisyonuna göre yerel çözümler sunarlar, her bir çözüm de atın kalitesi ile ilişkilendirilir.
- Yerel çözümlerin en iyileri iterasyon boyunca güncellenir.
- $\alpha$  ve  $\beta$  atları keşfedilen çözümlerden faydalanma (exploitation) işleminde,  $\delta$  ve  $\gamma$  atları ise keşif (exploration) sürecinde görev alır.

#### Algoritmaya Ait Parametrik Değerler

- At Popülasyonu Sayısı: 20
- İterasyon sayısı: 100
- Rastgele popülasyon oluşturma oranı: Öznitelik seçiminde Parato Law kuralları kullanılarak rastgelelik sağlanmıştır. Bu kurala göre bitler başlangıç aşamasında %80'ini 0, %20'si 1 olacak şekilde ayarlanmıştır.
- Eğitim ve Test Kümesi: Veri setlerinin %90'i eğitim, %10'u test için ayrılmıştır. Eğitim ve test kümesindeki veriler rastgele seçilmiştir.
- k-NN ve SVM sınıflandırıcı olarak kullanılmış ve sonuçlar 10 kat çapraz geçişleme ile elde edilmiştir.
- Uygunluk Fonksiyonu: Öznitelik seçimi probleminde ki uygunluk fonksiyonu, sadece sınıflandırma doğruluğuna değil aynı zamanda seçilen özniteliklerin sayısına da bağlıdır. Elde edilen iki öznitelik alt kümesinden doğruluk değeri daha yüksek olan tercih edilir. Ancak, iki alt küme aynı doğruluk değerine sahipse, içlerinden öznitelik sayısı daha az olan çözüm tercih edilir.
- Uygunluk fonksiyonunda kullanılan S katsayısı 0,0005 olarak belirlenmiştir.
- Durma Kriteri / Algoritmanın Sonlandırma Koşulu: Algoritmaya ait sonlandırma koşulu maksimum iterasyon sayısına ulaşmak olarak belirlenmiştir.

- Bilgisayar durumu: ilgili çalışmalarda belirtilen önerilere göre, Python 3.9 versiyonu ile kodlanarak 4GB RAM, Intel(R) Core(TM) i5-2450M 2.5 GHz CPU içeren bir bilgisayar kullanılarak çalıştırılmıştır.
- Eğitimi tamamlanan model 10 kat çapraz doğrulamada ayrılan test veri kümesi ile test edilip modelin doğruluğu ölçülür. Doğrulanan modele ait sınıflandırma başarısı değerlendirilerek akış tamamlanır.

### 3.6.2.9. Kullanılan veri setleri ve değerlendirme metrikleri

Geliştirilen algoritmanın değerlendirilmesi için UCI veri deposundan alınan farklı nitelik ve boyuttaki 20 farklı boyuttaki veri setleri kullanılmıştır. Bu veri setleri nitelik sayısına göre düşük, orta ve büyük boyutlu olmak üzere 3 farklı kategoride değerlendirilmiştir. İlgili veri setlerine ait özellikler Tablo 3.3'te belirtilmiştir.

**Tablo 3.3.** Kullanılan veri setleri.

Veri Seti Türü	Veri Seti Adı	Nitelik Sayısı	Örnek Sayısı	Sınıf Sayısı
Düşük Boyutlu Veri <=19	Iris	5	150	3
	Diabets	9	768	2
	Wine	13	178	3
	Hepatitis	19	155	2
	Vehicle	18	846	4
	Zoo	16	101	7
	Heart	13	270	2
	Breast Cancer	11	699	2
Orta Boyutlu Veri >20 ve <=90	Ionosphere	34	351	2
	Lung-Cancer	56	32	3
	Sonar	60	208	2
	BreastEW	30	569	2
	Movementlibras	90	360	15
	Soybean Small	35	47	4
	WaveformEW	40	5000	3
Büyük Boyutlu Veri >90	Arrhythmia	279	452	16
	Hillvalley	101	606	2
	Clean	166	476	2
	Semeion	256	1593	10
	CNAE	856	1080	9

Algoritmaya ait ortalama deęerlere ait denklemler ařaęıdaki gibidir.

### **Sınıflandırma doęruluęu ortalaması**

Algoritma N kez alıřtırıldıęında seilen zellik kumesi verilen sınıflandırıcının ne kadar doęru olduęunu gsteren bir gstergedir ve denklem 3.23'deki gibi hesaplanır.

$$AvgAcc = \frac{1}{N} \sum_{i=1}^N (AvgAcc^k) \quad (3.23)$$

### **Seilen zelliklerin ortalaması**

D veri seti boyutu olmak zere algoritma N kez alıřtırıldıęında seilen zelliklerin ortalamasının bir gstergesidir ve denklem 3.24'deki gibi hesaplanır.

$$AvSelectionAtt = \frac{1}{N} \sum_{i=1}^N \left( \frac{AvgAcc^k}{D} \right) \quad (3.24)$$

### **Ortalama uygunluk fonksiyonu**

Fit<sub>k</sub> uygunluk fonksiyonu olmak zere; Algoritma N kez alıřtırıldıęında elde edilen uygunluk fonksiyonunun ortalama deęerinin bir gstergesidir ve denklem 3.25'teki gibi hesaplanır.

$$mean = \frac{1}{N} \sum_{i=1}^N (fit_k) \quad (3.25)$$

Algoritmanın performans deęerlendirmesi iin literatrde eřitli istatistiksel yntemler bulunmaktadır. Tez alıřmasında bu yntemlerden Friedman ve Wilcoxon Testleri genel deęerlendirmeyi yapmak iin kullanılmıřtır.

### **Friedman ve wilcoxon testi**

Friedman Testi sıralı veriler veya normal daęılmayan bir aralıktaki lekli deęiřkenler iin kullanılan, parametrik olmayan ancak baęımlı gzlemlerde kullanılabilen bir yntemdir (McCrum-Gardner, 2008). Friedman testi algoritmaların karřılařtırılabilirlięini anlamak iin yaygın olarak kullanılmaktadır (Conover, 1999). İki'den fazla rneęin bulunduęu durumlarda kullanılan Friedman Testi, sınıflandırma bařarı metrięinin ortalama deęerlerini kullanarak deęerlendirme yapar.



H0: Karşılaştırılan algoritmalar birbirinden farklı değildir.

H1: Karşılaştırılan algoritmalar birbirinden farklıdır mantığı ile iki hipotez bulunmaktadır.

Sonucun doğru olabilmesi için alternatif hipotez H1'i ve sıfır hipotezi H0'ı belirlemek önemlidir. Sıfır hipotezi olarak bilinen iddia, sık sık karşılaştırılan algoritmalar arasında hiçbir fark olmadığını gösterir. Alternatif hipotez ise karşılaştırılan algoritmalar için birbirlerinin varyasyonları olduğunu ifade eder.

Literatürde kullanılan parametrik olmayan bir diğer istatistiksel test yöntemi ise Wilcoxon İşaretli Sıralar Testi'dir. Wilcoxon İşaretli Sıralar Testi önerilen algoritma için çalıştırılan veri setleri sonuçları kullanılarak, iyi bilinen algoritmalar ile kıyaslanmıştır. Kıyaslanacak iki algoritmanın veri setlerinin tamamı için sonuçları değerlendirilerek aralarındaki ilişki belirlenen güven değeri ile değerlendirilmiş ve anlamlı bir fark olup olmadığı belirlenmiştir. Bu tez kapsamında da Wilcoxon İşaret Testi uygulamasında literatürce en sık kullanılan 0,05 (p-değeri) güven seviyesi belirlenmiş ve deneysel çalışmalarda bu değer kullanılmıştır (Taheri & Hesamian, 2013). Test neticesinde hesaplanan p-değeri 0,05 güven seviyesinden büyük bir değer olarak hesaplandığında bu sonuç, değerlendirilen sonuçlar arasında istatistiksel açıdan bir farklılık olmadığı anlamına gelmektedir. Güven değeri 0,05'ten küçük hesaplandığında ise kıyaslanan algoritmalar arasında istatistiksel olarak anlamlı bir fark vardır sonucu ortaya çıkmaktadır. Bu fark kıyaslanan algoritmaların başarısı açısından değerlendirildiğinde anlamlı olarak birbirinden farklı sonucunu ortaya çıkarır.

Bu tez çalışmasında önerilen algoritmanın çeşitli veri setleri ile çalıştırıldığında elde edilen sonuçlarının iyi bilinen algoritmalar ile kıyaslanması amacıyla Friedman ve Wilcoxon Testleri kullanılmış ve algoritmanın genel performans değerlendirilmesi yapılmıştır.



#### **4. ARAŞTIRMA BULGULARI VE TARTIŞMA**

Bu bölümde nitelik seçimi problemi için geliştirilen İkili At Sürüsü Algoritması (BHOAFS) ile İkili Kaotik At Sürüsü Algoritmasının (BCHOAFS) performansı ölçülmüştür. Bölüm dört ana başlıktan oluşmaktadır. İlk olarak BHOAFS'nin k-NN (BHOAFS-k-NN) ve SVM(BHOAFS-SVM) sınıflandırıcıları ile olan performansları veri setleri üzerinde çalıştırılarak değerlendirilirken, ikinci bölümde daha iyi sonuçlar üreten BHOAFS-k-NN'in kararlılığı arttırmak amaçlı kaotik haritalar ile birleştirilerek önerilen BCHOAFS algoritmasının performansı değerlendirilmiştir. Üçüncü bölümde sonuçlar hem önerilen algoritmalar arasında hem de literatürce iyi bilinen algoritmalar ile kıyaslanmıştır. Dördüncü bölümde ise deneysel çalışmaların çeşitli istatistiksel testleri yapılarak geliştirilen yöntemin performansı doğrulanmıştır.

##### **4.1. BHOAFS'nin Performansı**

Sarmalayıcı tabanlı nitelik seçimi probleminde, seçilen özelliği değerlendirmek için bir sınıflandırıcıya ihtiyaç duyulur. Hangi özelliklerin seçileceği belirlendikten sonra sınıflandırma işlemi kullanılarak özelliklerin kullanılabilirliği ölçülür. Verinin özelliğine bağlı olarak tercih edilen sınıflandırma yöntemi seçimi, algoritmanın tahmin doğruluğunu arttırabilir ve tahmin için gereken hesaplama maliyetini azaltabilir. Uygun olarak seçilemeyen sınıflandırma algoritması tüm bu durumların aksine de neden olabilmektedir.

Veri madenciliği süreçleri ile makine öğreniminde yaygın olarak kullanılan k-NN ve SVM, en etkili sınıflandırıcı algoritmalarındandır (Wu vd., 2008). Yapılan çalışmalar incelendiğinde öznelik seçim algoritmalarında farklı veri tipleri için farklı sınıflandırıcılar kullanılmıştır (Albashish vd., 2021) (Sahran vd., 2018) (Sarafrazi & Nezamabadi-pour, 2013) (Silva vd., 2011)(Sun vd., 2012). Son yıllarda yapılan çalışmalarda ise veri kümesi iki sınıf içeriyorsa SVM'yi, ikiden fazla sınıf içeriyorsa k-NN sınıflandırıcısını tercih eden algoritmalar bulunmaktadır (Baş & Ülker, 2020). Önerilen yönteme ait deneysel çalışmalar kapsamında k-NN ve SVM sınıflandırıcıları ile ayrı ayrı çalıştırılarak performansının değerlendirildiği bu bölümde, iki farklı sınıflandırıcı kullanılarak önerilen BHOAFS algoritması için hangi sınıflandırıcının

daha iyi performans gösterdiği ölçülmüştür. Önerilen BHOAFS algoritması, k-NN ve SVM kullanılarak UCI deposundaki düşük, orta ve büyük ölçekli iyi bilinen veri kümeleri üzerinde ayrı ayrı test edilmiştir. Böylece BHOAFS'nin sınıflandırma performansı açısından en başarılı versiyonu belirlenmeye çalışılmıştır.

BHOAFS, seçilen veri kümeleri için ön işleme adımıyla başlar. İlgili veriler uygun hale getirildikten sonra eğitim seti ve test seti olmak üzere iki kısma ayrılır ve orijinal veri setinden özniteliklerin bir alt kümesi oluşturulur. Oluşturulan alt küme kullanılarak belirlenen iterasyon sayısınca algoritma, seçilen sınıflandırma yöntemi ile uygunluk fonksiyonu hesaplanarak değerlendirme işlemine devam eder. Eğitim veri kümesi tarafından oluşturularak kullanılan alt küme boyutu test veri kümesinin de alt kümesi olarak kullanılır. İndirgenmiş test seti de kullanılan sınıflandırma algoritması ile sınıflandırılır ve sonuçlar değerlendirilir. Algoritmanın durdurma kriteri karşılandığında en uygun özellikler kabul edilir ve süreç tamamlanır.

Önerilen algoritma BHOAFS Tablo 4.1'de öznitelik seçimi yapmadan ve öznitelik seçimi yapılarak elde edilen sınıflandırma doğruluğu k-NN ve SVM için ayrı ayrı görülmektedir. UCI deposundan alınan 20 veri seti için uygulama çalıştırılmıştır. BHOAFS sonuçlarına göre veri kümelerinin çoğu için hem k-NN hem de SVM için benzer sonuçlar vermiştir. k-NN ile sınıflandırılan BHOAFS algoritmasının Iris, Diabetes, Wine, Vehicle, Heart, Soybean Small, Sonar, BreastEW, Movementlibras, Hillvalley, Clean ve Semeion veri setlerinde daha iyi sonuçlar verdiği görülmüştür. Hepatit, Zoo, Ionosphere, WaveformEW ve Arrhythmia veri kümeleri için SVM ile sınıflandırılan BHOAFS'nin daha iyi performans gösterdiği gözlenmiştir. Breast Cancer , Lung Canceri ve CNAE veri kümeleri için hem k-NN hem de SVM sınıflandırıcıları aynı sonuçları vermiştir. Deneysel sonuçlar k-NN'nin sınıflandırma performansının SVM'den üstün olduğunu göstermiştir.

Tablo 4.2'de önerilen yöntemin iyi bilinen metasezgisel algoritmalar ile ortalama uygunluk fonksiyonu karşılaştırması görülmektedir. BHOAFS-k-NN için Wine, Vehicle, Heart, Soybean Small, Lung-Cancer, Sonar, BreastEW, Movementlibras, Hillvalley, Clean ve CNAE veri kümeleri daha düşük uygunluk sonuçları üretmiştir. BHOAFS-SVM, Hepatit, Zoo, Breast Cancer, Ionosphere, WaveformEW, Arrhythmia ve Semeion veri kümeleri için daha düşük uygunluk sonuçları üretmiştir. Elde edilen sonuçlara göre k-NN sınıflandırma algoritması büyük ölçekli veriler için daha performanslı sonuçlar üretmektedir. Bahsedilen performans ölçüsü daha yüksek bir

doğruluk deęerine ve daha düşük bir nitelik seçimi sayısına sahip olma durumudur. Bu durum denklem 3.19'a göre hesaplanan uygunluk deęerlerini gösteren Tablo 4.2'de açıkça görölmektedir.

**Tablo 4.1.** BHOAFS'nin k-NN ve SVM sınıflandırıcılarına ait doğruluk ve öznelik karşılaştırması.

Benchmark	Nitelik	Öznelik Seçimi Yapmadan		BHOAFS-kNN		BHOFS-SVM	
		Doğruluk k-NN	Doğruluk SVM	Doğruluk	Öznelik	Doğruluk	Öznelik
Iris	5	0.97333	0.97333	<b>0.98000</b>	<b>3</b>	0.97333	<b>3</b>
Diabetes	9	0.72857	0.72857	0.77662	<b>5</b>	<b>0.77623</b>	<b>5</b>
Wine	13	0.75000	0.75000	<b>0.95556</b>	<b>5</b>	0.93889	7
Hepatitis	19	0.63333	0.63333	0.74000	<b>5</b>	<b>0.76000</b>	6
Vehicle	18	0.65647	0.65647	<b>0.75529</b>	<b>4</b>	0.65059	6
Zoo	16	0.87272	0.87272	0.94545	7	<b>0.96364</b>	<b>6</b>
Heart	13	0.64814	0.64814	<b>0.85926</b>	<b>3</b>	0.85556	6
Breast Cancer	11	0.61000	0.61000	<b>0.97857</b>	<b>6</b>	<b>0.97857</b>	7
Ionosphere	34	0.81944	0.81944	0.93056	<b>6</b>	<b>0.96944</b>	12
Lung-Cancer	56	0.80000	0.80000	<b>0.97500</b>	13	<b>0.97500</b>	<b>11</b>
Sonar	60	0.83809	0.83809	<b>0.94762</b>	21	0.92857	<b>17</b>
BreastEW	30	0.92280	0.92280	<b>0.96140</b>	11	0.95611	<b>2</b>
Movementlibras	90	0.77777	0.77777	<b>0.83611</b>	27	0.83333	<b>26</b>
WaveformEW	40	0.80540	0.80540	0.84580	<b>18</b>	<b>0.87520</b>	26
Arrhythmia	279	0.60217	0.60217	0.70000	138	<b>0.70870</b>	<b>83</b>
Hillvalley	101	0.54426	0.54426	<b>0.65574</b>	<b>14</b>	0.48033	25
Soybean Small	35	0.96000	0.96000	<b>1.00000</b>	<b>15</b>	0.80000	23
Clean	166	0.30208	0.30208	<b>0.95417</b>	131	0.93958	<b>102</b>
Semeion	256	0.96812	0.96812	<b>0.98188</b>	<b>61</b>	0.98063	117
CNAE	856	0.99814	0.99814	<b>0.99815</b>	663	<b>0.99815</b>	<b>661</b>

**Tablo 4.2.** BHOAFS'nin literatürdeki diğer algoritmalar ile ortalama uygunluk fonksiyonu karşılaştırması.

Benchmark	BHOAFS-k-NN	BHOAFS-SVM	GA	PSO	ALO	GWO	SSA
Wine	<b>0.05015</b>	0.06512	0.23000	0.54000	0.37200	0.43000	0.23000
Hepatitis	0.26477	<b>0.24444</b>	1.60000	1.50000	1.29300	2.19500	1.26300
Vehicle	<b>0.25004</b>	0.35258	2.30000	2.10000	3.01900	2.94500	2.39300
Zoo	0.05963	<b>0.04225</b>	1.16000	2.00000	1.87500	1.02700	0.75800
Heart	<b>0.14703</b>	0.14838	1.25000	1.85000	2.02600	1.30100	1.34800
Breast Cancer	0.02576	<b>0.02485</b>	0.38000	0.37000	0.42500	0.41500	0.39900
Ionosphere	0.07698	<b>0.03672</b>	1.27000	1.76000	1.19900	1.41500	1.21400
Lung Cancer	<b>0.03243</b>	0.03279	1.50000	1.72400	1.90200	2.84400	1.29600
Sonar	<b>0.05836</b>	0.07788	1.69000	3.19000	1.76900	2.25600	0.06900
BreastEW	<b>0.04454</b>	0.05278	0.42000	0.54000	0.56400	0.65400	0.51200
Soybean Small	<b>0.00571</b>	0.20143	0.60000	0.40000	0.01900	2.91300	0.04900
Movementlibras	<b>0.16925</b>	0.17211	2.96000	3.33000	3.88900	2.78400	1.74900
Waveform	0.15816	<b>0.12705</b>	2.18000	2.40000	2.68100	2.64800	2.02000
Arrhythmia	0.30205	<b>0.29542</b>	0.60000	0.90000	0.37400	0.36300	2.93500
Semeion	0.02556	<b>0.02461</b>	–	–	–	–	7.28600
Clean	<b>0.04748</b>	0.06367	0.64000	0.92000	0.16100	0.18400	0.96500
CNAE	<b>0.00409</b>	0.00411	2.67000	3.27000	1.66200	2.00000	2.06900
Hillvalley	<b>0.34943</b>	0.52200	2.45000	2.63000	4.05200	4.08800	2.92000

## 4.2. BCHOAFS'nin Performansı

Kaotik haritalar, başlangıç koşullarına duyarlı ve düzensiz/rasgele davranışa göre değişkenlik gösteren deterministik sistemlerdir. Kaos teorisi ise deterministik kuralları ve dinamik sistem davranışlarını inceler (Zawbaa vd., 2016). Başlangıç koşullarına göre, kaotik değişkenler belirli aralıklarla tüm durumları tekrarlamadan geçebilirler. Kaotik sayı dizilerini yapılandırarak oluşturmak hızlı ve kolaydır. Bu davranışlardan dolayı, kaos araması tepeye tırmanma ve yerel optimumdan kaçma durumları için rastgele aramaya göre daha iyidir. Bu nedenle, denklemlerde kullanılan temel bir hareket modunu temsilen veya stokastik değişkenleri güncellemek amacıyla kaos haritaları birçok optimizasyon algoritmasında uygulanmıştır (Abdel-Raouf vd., 2014).

Önerilen algoritmada, BHOAFS'nin öznelik seçim performansını iyileştirmek, arama uzayında etkin çözümler bulmak ve algoritmanın kararlılığını artırmak için arama sürecinin başında rastgele sayı üretici olarak tek boyutlu kaotik haritalar kullanılmaktadır. Yapılan deneysel çalışmada Logistic, Tent, Piecewise, Singer ve Sinusoidal olmak üzere beş adet kaotik harita kullanılmıştır. Bu haritalara ait matematiksel denklemler Tablo 3.1'de gösterilmiştir.

Bu bölümde önerilen beş kaotik iyileştiricinin performansı Lojistik BHOAFS (BCHOAFS1), Tent BHOAFS (BCHOAFS2), Piecewise BHOAFS (BCHOAFS3), Singer BHOAFS (BCHOAFS4), Sinusoidal BHOAFS (BCHOAFS5) olmak üzere 18 veri seti için ölçülmüştür. Bölüm 4.1'de bahsedildiği üzere deneysel sonuçlar k-NN'nin sınıflandırma performansının SVM'den üstün olduğunu göstermiştir. Bu sebeple BHOAFS-k-NN üzerinde kaotik haritalar uygulanmıştır. Önerilen beş algoritma ayrı ayrı benzer başlangıç koşulları ile çalıştırılmıştır. Önerilen algoritma için seçilen tüm kaotik haritalarda başlangıç değeri 0.7 olarak ayarlanmıştır (Sayed, Hassanien, vd., 2019b). Tablo 4.3 önerilen BCHOAFS'nin (yani, BCHOAFS1, BCHOAFS2, BCHOAFS3, BCHOAFS4, BCHOAFS5) tüm iterasyonlar sırasında elde edilen en iyi doğruluk değerlerini göstermektedir. Veri kümelerinin yaklaşık %70'inde, BCHOAFS3 ve BCHOAFS4, BHOAFS-k-NN'den ya daha iyi ya da aynı sonuçları vermiştir. Tablo 4.4 ise BCHOAFS sürümlerinin seçilen özellik sayıları hakkında bilgi vermektedir. BCHOAFS versiyonlarının ortalama doğruluk ve seçilen özelliklerin sayısı Tablo 4.5'te görülebilir. Tablo 4.1'den Tablo 4.5'e kadar en iyi sonuçlar koyu renk ile belirtilmiştir.



Kaotik haritalar elde edilen doğruluk ve seçilen özellik sayısına göre sıralandığında en iyi sonuçları Piecewise ve Singer kaotik haritaları vermiştir. Benzer şekilde Lojistik, Tent ve Sinusoidal haritalarda yakın sonuçlar üretmiştir. Tüm bu sonuçlar, Piecewise ve Singer Kaotik Harita tabanlı BCHOAFS'nin (BCHOAFS3 ve BCHOAFS4), önerilen BCHOAFS sürümlerine kıyasla kaliteyi artırırken seçilen özelliklerin sayısını azalttığını kanıtlamaktadır.

**Tablo 4.3.** BHOAFS k-NN sınıflandırıcısına ait kaotik varyasyonların doğruluk değerleri.

Benchmark	BCHOAFS1	BCHOAFS2	BCHOAFS3	BCHOAFS4	BCHOAFS5
Wine	<b>0.95556</b>	<b>0.95556</b>	<b>0.95556</b>	<b>0.95556</b>	<b>0.95556</b>
Hepatitis	0.72667	0.72667	0.72667	<b>0.74000</b>	0.72667
Vehicle	0.73882	<b>0.75765</b>	<b>0.75765</b>	0.75529	<b>0.75765</b>
Zoo	<b>0.96364</b>	<b>0.96364</b>	<b>0.96364</b>	0.94545	<b>0.96364</b>
Heart	<b>0.86296</b>	<b>0.86296</b>	<b>0.86296</b>	0.85926	<b>0.86296</b>
Breast Cancer	<b>0.97857</b>	<b>0.97857</b>	<b>0.97857</b>	<b>0.97857</b>	<b>0.97857</b>
Ionosphere	<b>0.93056</b>	<b>0.93056</b>	0.92778	0.92500	0.92222
Lung-Cancer	<b>0.97500</b>	<b>0.97500</b>	<b>0.97500</b>	<b>0.97500</b>	<b>0.97500</b>
Sonar	<b>0.93810</b>	0.93333	<b>0.93810</b>	0.92857	<b>0.93810</b>
BreastEW	<b>0.96140</b>	<b>0.96140</b>	<b>0.96140</b>	<b>0.96140</b>	<b>0.96140</b>
Movementlibras	0.83611	0.83333	<b>0.84167</b>	0.82778	0.83333
WaveformEW	0.84680	0.84540	0.84580	0.84840	<b>0.84880</b>
Arrhythmia	<b>0.73478</b>	0.70217	0.70435	0.70217	0.69783
Hillvalley	0.64098	0.63607	0.63443	0.65082	<b>0.65410</b>
Soybean Small	<b>1.00000</b>	<b>1.00000</b>	<b>1.00000</b>	<b>1.00000</b>	<b>1.00000</b>
Clean	<b>0.94792</b>	0.93542	0.94375	0.94375	0.93958
Semeion	0.98000	0.98000	<b>0.98063</b>	0.97938	0.97938
CNAE	<b>0.99815</b>	<b>0.99815</b>	<b>0.99815</b>	<b>0.99815</b>	<b>0.99815</b>

**Tablo 4.4.** BHOAFS k-NN sınıflandırıcısına ait kaotik varyasyonların öznelilik sayıları.

Benchmark	BCHOAFS1	BCHOAFS2	BCHOAFS3	BCHOAFS4	BCHOAFS5
Wine	<b>5</b>	6	<b>5</b>	<b>5</b>	<b>5</b>
Hepatitis	8	5	<b>4</b>	5	8
Vehicle	11	<b>4</b>	<b>4</b>	<b>4</b>	<b>4</b>
Zoo	<b>6</b>	<b>6</b>	<b>6</b>	7	<b>6</b>
Heart	4	4	4	<b>3</b>	4
Breast Cancer	<b>6</b>	<b>6</b>	7	<b>6</b>	<b>6</b>
Ionosphere	<b>6</b>	7	<b>6</b>	<b>6</b>	9
Lung-Cancer	<b>15</b>	17	16	16	<b>15</b>
Sonar	<b>21</b>	28	32	26	29
BreastEW	7	14	12	12	<b>5</b>
Movementlibras	39	<b>33</b>	46	50	<b>33</b>
WaveformEW	21	<b>15</b>	18	22	18
Arrhythmia	160	177	158	150	<b>142</b>
Hillvalley	28	28	49	<b>21</b>	24
Soybean Small	17	<b>13</b>	15	19	16
Clean	<b>107</b>	123	120	130	135
Semeion	87	93	<b>58</b>	59	82
CNAE	666	<b>659</b>	676	674	676

**Tablo 4.5.** BHOAFS k-NN sınıflandırıcısına ait kaotik varyasyonların ortalama doğruluk ve öznelilik sayısı.

Benchmark	BCHOAFS1		BCHOAFS2		BCHOAFS3		BCHOAFS4		BCHOAFS5	
	Doğruluk	Nitelik	Doğruluk	Nitelik	Doğruluk	Nitelik	Doğruluk	Nitelik	Doğruluk	Nitelik
Wine	<b>0.95556</b>	6.0	<b>0.95556</b>	6.2	0.95500	6.0	<b>0.95556</b>	<b>5.8</b>	0.95500	6.1
Hepatitis	0.71933	8.6	<b>0.72333</b>	7.8	0.71867	<b>7.6</b>	0.72067	7.7	0.70733	<b>7.6</b>
Vehicle	0.72954	10.3	0.73753	<b>8.6</b>	0.73612	9.0	0.73447	9.5	<b>0.73965</b>	9.0
Zoo	<b>0.94727</b>	7.9	0.93273	8.7	0.94364	8.9	0.93273	<b>7.8</b>	0.94182	9.2
Heart	0.86111	4.5	<b>0.86185</b>	4.3	<b>0.86185</b>	4.3	0.85926	<b>3.4</b>	0.86111	4.5
Breast Cancer	<b>0.97857</b>	6.6	<b>0.97857</b>	<b>6.0</b>	<b>0.97857</b>	7.0	0.97814	<b>6.0</b>	<b>0.97857</b>	6.2
Ionosphere	<b>0.91972</b>	7.0	0.91528	8.2	0.91806	8.1	0.91472	<b>6.8</b>	0.91444	8.4
Lung-Cancer	0.97250	24.7	<b>0.97500</b>	24.2	<b>0.97500</b>	22.6	<b>0.97500</b>	24.0	0.97000	<b>19.9</b>
Sonar	<b>0.92190</b>	<b>25.1</b>	0.91714	25.2	0.91952	28.0	0.91667	28.1	0.91667	27.7
BreastEW	0.96088	14.8	0.96070	15.8	<b>0.96140</b>	<b>14.1</b>	0.96070	15.0	0.96053	15.2
Movementlibras	<b>0.82472</b>	36.4	0.82222	<b>32.7</b>	0.82056	35.9	0.82056	35.6	0.82139	39.6
WaveformEW	0.84320	19.3	0.84322	20.1	0.84240	<b>18.7</b>	0.84220	22.5	<b>0.84378</b>	19.2
Arrhythmia	0.68783	155.1	0.68761	157.8	<b>0.69022</b>	158.3	0.68783	155.3	0.68630	<b>154.2</b>
Hillvalley	0.62721	<b>27.1</b>	0.62525	30.5	0.62492	37.7	0.62918	39.0	<b>0.63098</b>	29.1
Soybean Small	<b>1.00000</b>	23.2	<b>1.00000</b>	22.9	<b>1.00000</b>	22.8	<b>1.00000</b>	22.2	<b>1.00000</b>	<b>22.0</b>
Clean	<b>0.94062</b>	114.5	0.92813	109.2	0.93125	<b>106.9</b>	0.93354	108.5	0.93042	108.0
Semeion	0.97606	91.6	<b>0.97625</b>	85.5	0.97550	91.9	0.97594	<b>77.1</b>	0.97506	103.7
CNAE	<b>0.99815</b>	687.1	<b>0.99815</b>	<b>682.3</b>	<b>0.99815</b>	687.7	<b>0.99815</b>	688.1	0.99815	684.6

### 4.3. Geliştirilen Yöntemin Literatürdeki Algoritmalar ile Performans Kıyaslaması

Önceki bölümde önerilen BHOAFS-k-NN kaotik haritalar ile birleştirildiğinde elde edilen sonuçlar gösterilmiştir. Bu bölümde ise önerilen algoritmaların literatürde iyi bilinen algoritmalar ile karşılaştırılması yapılmıştır.

Tablo 4.6'da, hem k-NN hem de SVM sınıflandırıcıları için önerilen yöntemin ortalama doğruluk değerine ait literatürdeki iyi bilinen optimizasyon yöntemleriyle (GA, PSO, ALO, GWO, SSA) karşılaştırılması gösterilmiştir (Arora & Anand, 2019), (Baş & Ülker, 2020). Tablo 4.6'dan Vehicle, Heart, Breast Cancer, Soya Small, Lung-Cancer, Sonar, BreastEW, Movementlibras, Clean, Semeion, CNAE veri kümeleri için BHOAFS-k-NN'nin, karşılaştırılan diğer algoritmalar arasında iyi sonuçlar verdiği gözlenmektedir. BHOAFS-SVM'nin ise Breast Cancer, Ionosphere, Lung Cancer, WaveformEW, Arrhythmia ve CNAE veri kümeleri için en iyi sonuçlar verdiği görülmektedir.

Tablo 4.7, önerilen algoritmanın literatürdeki iyi bilinen ikili optimizasyon yöntemleriyle (yani, BGA, BPSO, BALO, BGWO, BSSA) karşılaştırmasını göstermektedir. Tablo 4.7'de görüldüğü üzere, Vehicle, Heart, Breast Cancer, Soybean Small, Lung Cancer, Sonar, BreastEW, Movementlibras, Clean, Semeion ve CNAE veri setleri için BHOAFS-k-NN, karşılaştırılan diğer algoritmalar arasında iyi sonuçlar vermiştir. BHOAFS-SVM, Breast Cancer, Ionosphere, Lung Cancer, WaveformEW, Arrhythmia ve CNAE veri kümeleri için en iyi sonuçları vermiştir. Deneysel sonuçlar, k-NN (BHOAFS-k-NN) tarafından sınıflandırılan BHOAFS'nin, SVM (BHOAFS-SVM) tarafından sınıflandırılan BHOAFS versiyonundan daha iyi performans gösterdiğini kanıtlamıştır.

Tablo 4.8'de, algoritmanın seçilen özelliklerinin karşılaştırması gösterilirken BHOAFS-k-NN, Wine, Hepatitis, Vehicle, Heart, Ionosphere, WaveformEW, Hillvalley ve Semeion veri kümeleri için karşılaştırılan diğer algoritmalar arasında en az özellik seçimine sahiptir. Zoo, Lung Cancer, Sonar, BreastEW, Movementlibras ve Arrhythmia veri setleri için BHOAFS-SVM en az özellik seçimini yapmıştır.

k-NN ve SVM için önerilen algoritmanın çalışma süresi Tablo 4.9'da gösterilmiştir. Tablo 4.9'da belirtilen çalışma süreleri algoritmanın en iyi çözümü ararken kullandığı çalışma süresidir. İlgili veri setlerinin öznelik seçimi yapmadan doğrudan k-NN ve SVM sınıflandırıcıları tarafından sınıflandırıldıkları çalışma süresi Tablo 4.9'dan çok daha kısadır. Tablolardaki en iyi sonuçlar koyu renk ile ifade edilmiştir. Doğruluk değeri artarken öznelik sayısının azalması hedeflenen algoritma üzerinde amaçlanan bir durumdur. Bu durum önerilen algoritmanın çok amaçlı olduğunu göstermektedir.

**Tablo 4.6.** BHOAFS'nin literatürdeki diğer optimizasyon algoritmaları ile ortalama doğruluk karşılaştırması.

Benchmark	BHOAFS-k-NN	BHOAFS-SVM	GA	PSO	ALO	GWO	SSA
Wine	0.95556	0.93889	0.95700	0.92300	0.94200	0.91200	<b>0.97500</b>
Hepatitis	0.74000	0.76000	0.87500	0.86050	0.88250	0.84790	<b>0.90000</b>
Vehicle	<b>0.75529</b>	0.65059	0.71630	0.66950	0.68140	0.61720	0.72010
Zoo	0.94545	0.96364	0.85400	0.82400	0.80500	0.87500	<b>0.96750</b>
Heart	<b>0.85926</b>	0.85556	0.82400	0.82200	0.80200	0.80700	0.85210
Breast Cancer	<b>0.97857</b>	<b>0.97857</b>	0.95500	0.95100	0.95000	0.95300	0.96850
Ionosphere	0.93056	<b>0.96944</b>	0.82400	0.84800	0.84300	0.81900	0.91000
Lung Cancer	<b>0.97500</b>	<b>0.97500</b>	0.48200	0.56270	0.50560	0.50140	0.90830
Sonar	<b>0.94762</b>	0.92857	0.71700	0.72300	0.71400	0.71400	0.93750
BreastEW	<b>0.96140</b>	0.95611	0.93500	0.94900	0.94200	0.94900	0.95200
Soybean Small	<b>1.00000</b>	0.80000	0.94380	0.86480	0.90980	0.92050	<b>1.00000</b>
Movementlibras	<b>0.83611</b>	0.83333	0.69020	0.64660	0.65970	0.68660	0.82220
Waveform	0.84580	<b>0.87520</b>	0.76200	0.76200	0.76900	0.76500	0.80080
Arrhythmia	0.70000	<b>0.70870</b>	0.58020	0.57070	0.54620	0.56410	0.68280
Semeion	<b>0.98188</b>	0.98063					0.20750
Clean	<b>0.95417</b>	0.93958	0.76480	0.77840	0.80980	0.79530	0.90680
CNAE	<b>0.99815</b>	<b>0.99815</b>	0.82460	0.81470	0.79620	0.84070	0.79000
Hillvalley	<b>0.65574</b>	0.48033	0.56270	0.55070	0.57090	0.55440	0.66580

**Tablo 4.7.** BHOAFS'nin literatürdeki diğer ikili optimizasyon algoritmaları ile ortalama doğruluk karşılaştırması.

Benchmark	BHOAFS-k-NN	BHOAFS-SVM	BGA	BPSO	BALO	BGWO	BSSA
Wine	0.95556	0.93889	0.95700	0.92300	0.94200	0.91200	<b>0.97500</b>
Hepatitis	0.74000	0.76000	0.87500	0.86050	0.88250	0.84790	<b>0.90000</b>
Vehicle	<b>0.75529</b>	0.65059	0.71630	0.66950	0.68140	0.61720	0.72010
Zoo	0.94545	0.96364	0.85400	0.82400	0.80500	0.87500	<b>0.96750</b>
Heart	<b>0.85926</b>	0.85556	0.82400	0.82200	0.80200	0.80700	0.85210
Breast Cancer	<b>0.97857</b>	<b>0.97857</b>	0.95500	0.95100	0.95000	0.95300	0.96850
Ionosphere	0.93056	<b>0.96944</b>	0.82400	0.84800	0.84300	0.81900	0.91000
Lung Cancer	<b>0.97500</b>	<b>0.97500</b>	0.48200	0.56270	0.50560	0.50140	0.90830
Sonar	<b>0.94762</b>	0.92857	0.71700	0.72300	0.71400	0.71400	0.93750
BreastEW	<b>0.96140</b>	0.95611	0.93500	0.94900	0.94200	0.94900	0.95200
Soybean Small	<b>1.00000</b>	0.80000	0.94380	0.86480	0.90980	0.92050	0.10000
Movementlibras	<b>0.83611</b>	0.83333	0.69020	0.64660	0.65970	0.68660	0.82220
Waveform	0.84580	<b>0.87520</b>	0.76200	0.76200	0.76900	0.76500	0.80080
Arrhythmia	0.70000	<b>0.70870</b>	0.58020	0.57070	0.54620	0.56410	0.68280
Semeion	<b>0.98188</b>	0.98063					0.20750
Clean	<b>0.95417</b>	0.93958	0.76480	0.77840	0.80980	0.79530	0.90680
CNAE	<b>0.99815</b>	<b>0.99815</b>	0.82460	0.81470	0.79620	0.84070	0.79000
Hillvalley	0.65574	0.48033	0.56270	0.55070	0.57090	0.55440	<b>0.66580</b>



**Tablo 4.8.** BHOAFS'nin literatürdeki diğer ikili optimizasyon algoritmaları ile öznitelik sayısı karşılaştırması.

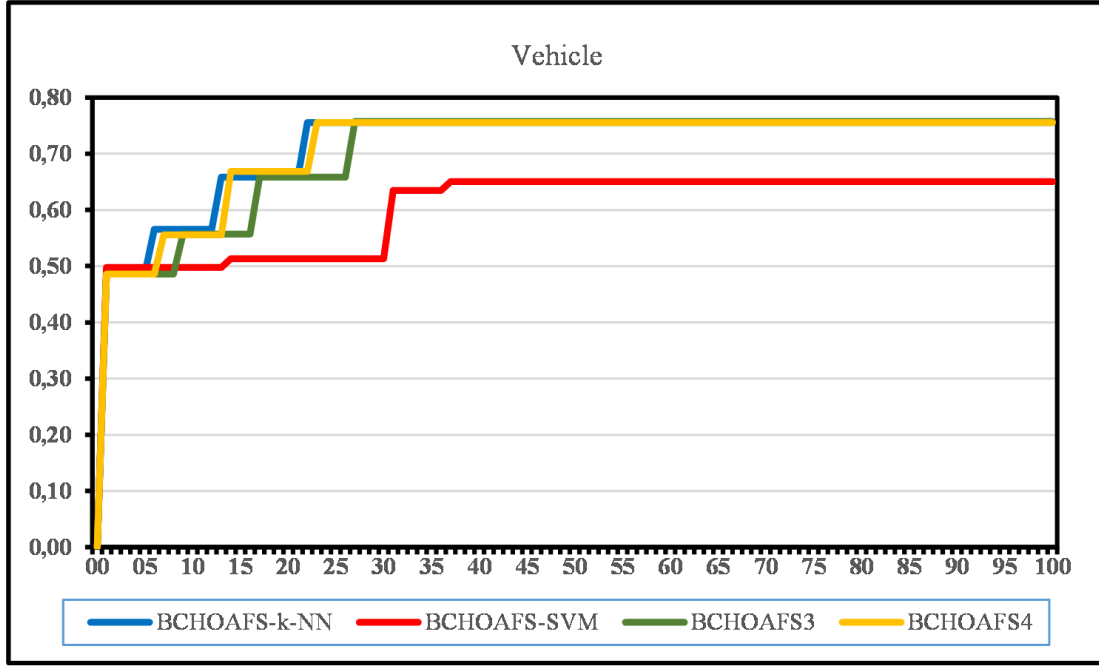
Benchmark	BHOAFS-k-NN	BHOAFS-SVM	BSSA	BBA	BGWO	BPSO
Wine	<b>5</b>	7	<b>5</b>	<b>5</b>	6	<b>5</b>
Hepatitis	<b>5</b>	6	9	7	8	7
Vehicle	<b>4</b>	6	8	9	11	8
Zoo	7	<b>6</b>	<b>6</b>	<b>6</b>	<b>6</b>	7
Heart	<b>3</b>	6	5	6	6	5
Breast Cancer	6	7	<b>4</b>	<b>4</b>	<b>4</b>	<b>4</b>
Ionosphere	<b>6</b>	12	14	17	15	14
Lung Cancer	13	<b>11</b>	24	28	25	28
Sonar	21	<b>17</b>	28	27	28	29
BreastEW	11	<b>2</b>	12	13	20	17
Soybean Small	15	23	<b>14</b>	<b>14</b>	18	16
Movementlibras	27	<b>26</b>	40	39	54	40
Waveform	<b>18</b>	26	23	22	30	23
Arrhythmia	138	<b>83</b>	147	146	144	142
Semeion	<b>61</b>	117	137	133	152	134
Clean	131	102	87	87	111	<b>86</b>
CNAE	663	661	<b>421</b>	428	482	433
Hillvalley	<b>14</b>	25	49	49	65	51

**Tablo 4.9.**BHOAFS'nin k-NN ve SVM sınıflandırmaları için CPU süresi (saniye).

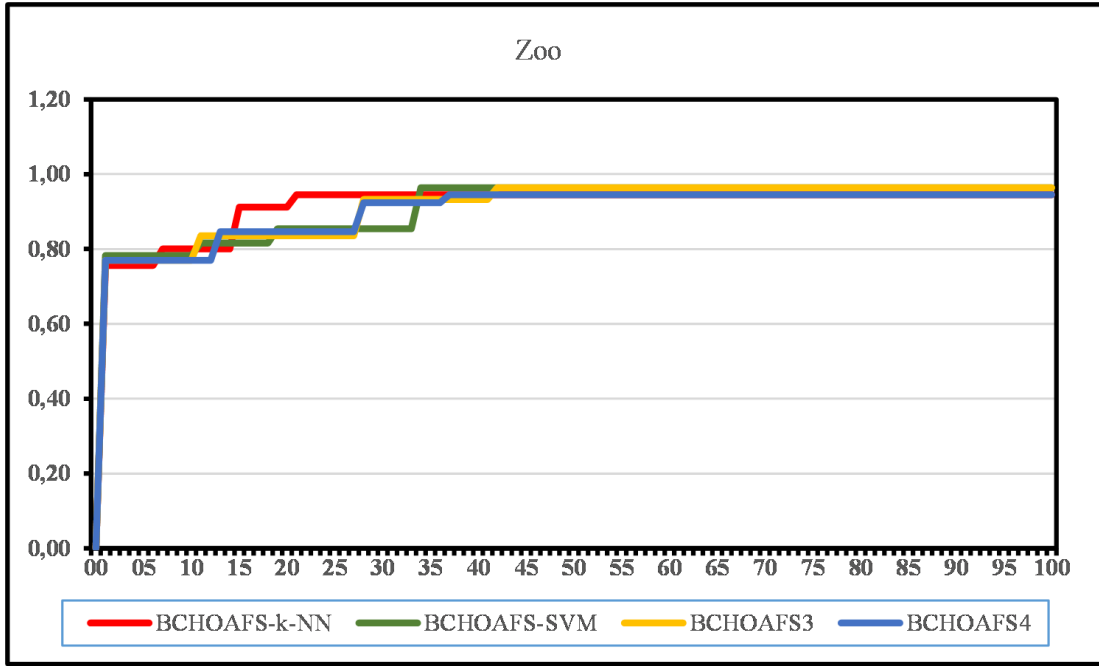
Benchmark	BHOAFS-k-NN	BHOAFS-SVM
Wine	7497	<b>6754</b>
Hepatitis	18369	<b>10529</b>
Vehicle	<b>20664</b>	83831
Zoo	14313	<b>11144</b>
Heart	10845	<b>4316</b>
Breast Cancer	8938	<b>3556</b>
Ionosphere	52329	<b>36091</b>
Lung-Cancer	<b>19413</b>	35866
Sonar	92465	<b>30284</b>
BreastEW	<b>19791</b>	233658
Movementlibras	<b>79219</b>	103651
WaveformEW	<b>514219</b>	2086982
Arrhythmia	<b>355225</b>	558137
Hillvalley	126082	<b>98821</b>
Soybean Small	10245	<b>7271</b>
Clean	<b>222031</b>	385066
Semeion	649699	<b>548796</b>
CNAE	1565015	<b>1307388</b>

#### 4.4. Yakınsama Grafikleri

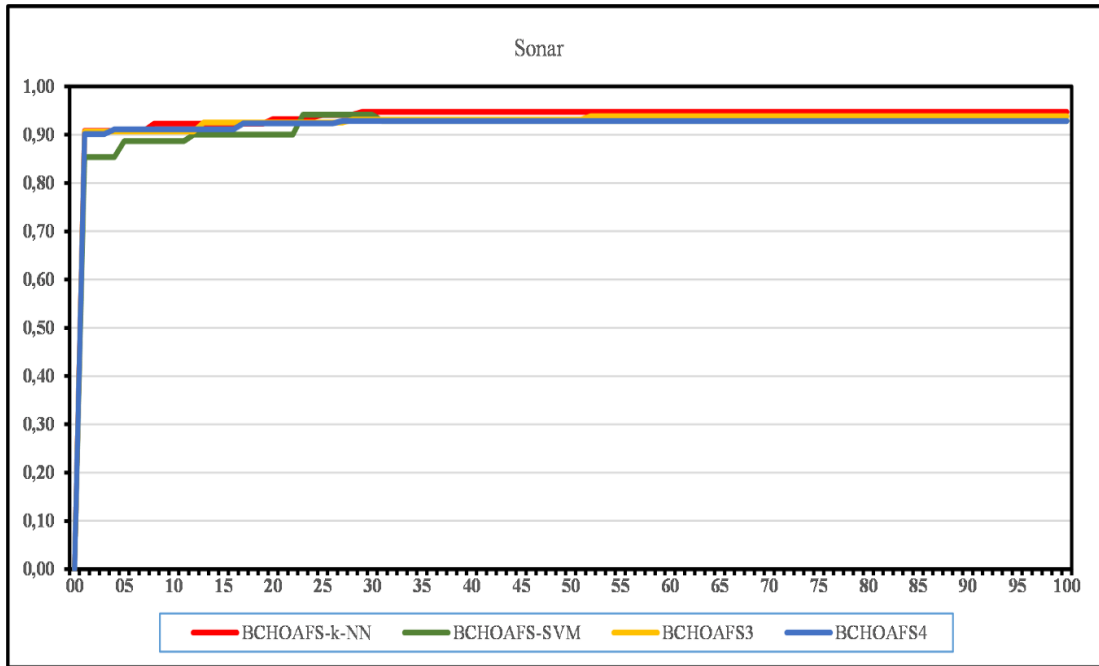
Bu bölümde önerilen algoritmalarından BHOAFS-k-NN, BHOAFS-SVM, BHOAFS ve BHOAFS için her veri seti için yakınsama grafiklerinin gösterilmiştir.



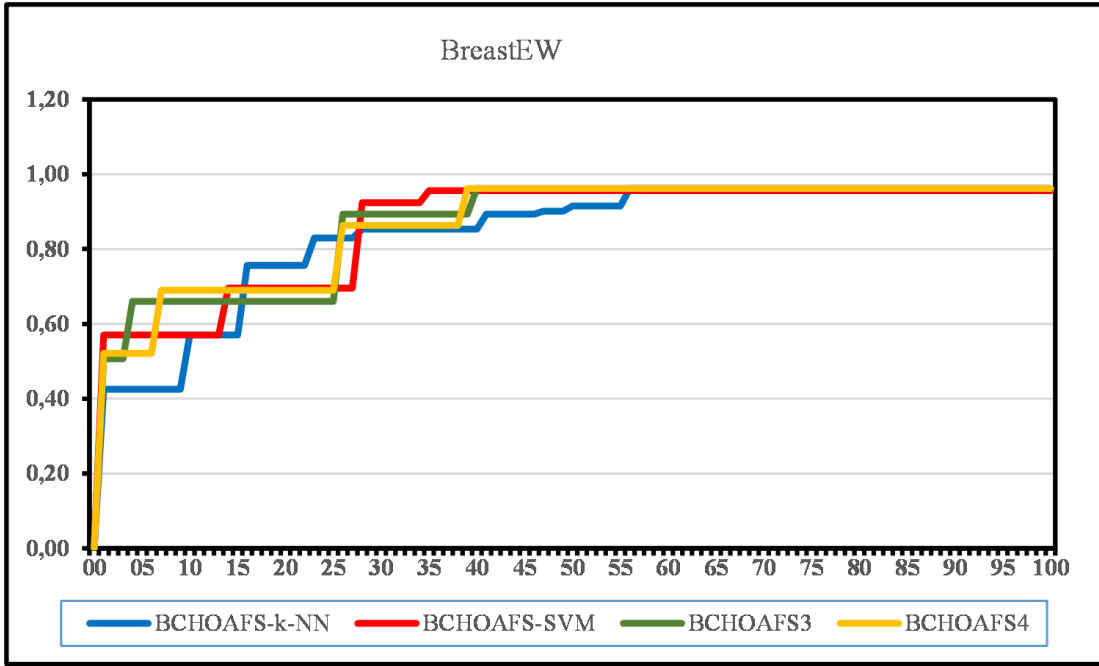
Şekil 4.1. Vehicle veri seti için yakınsama grafiği.



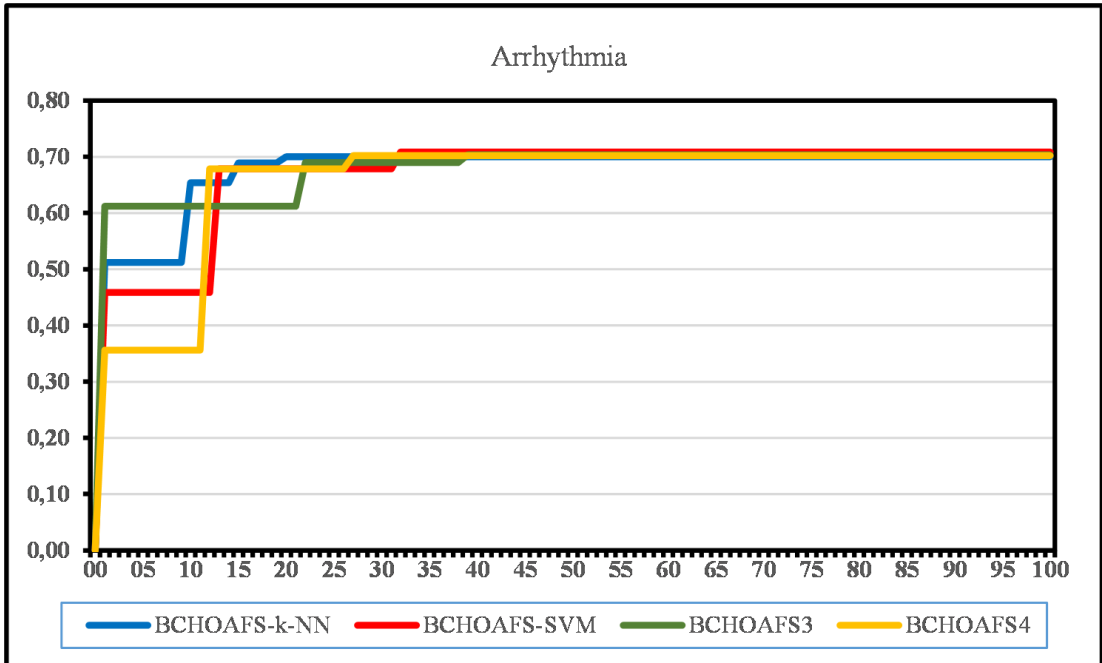
Şekil 4.2. Zoo veri seti için yakınsama grafiği.



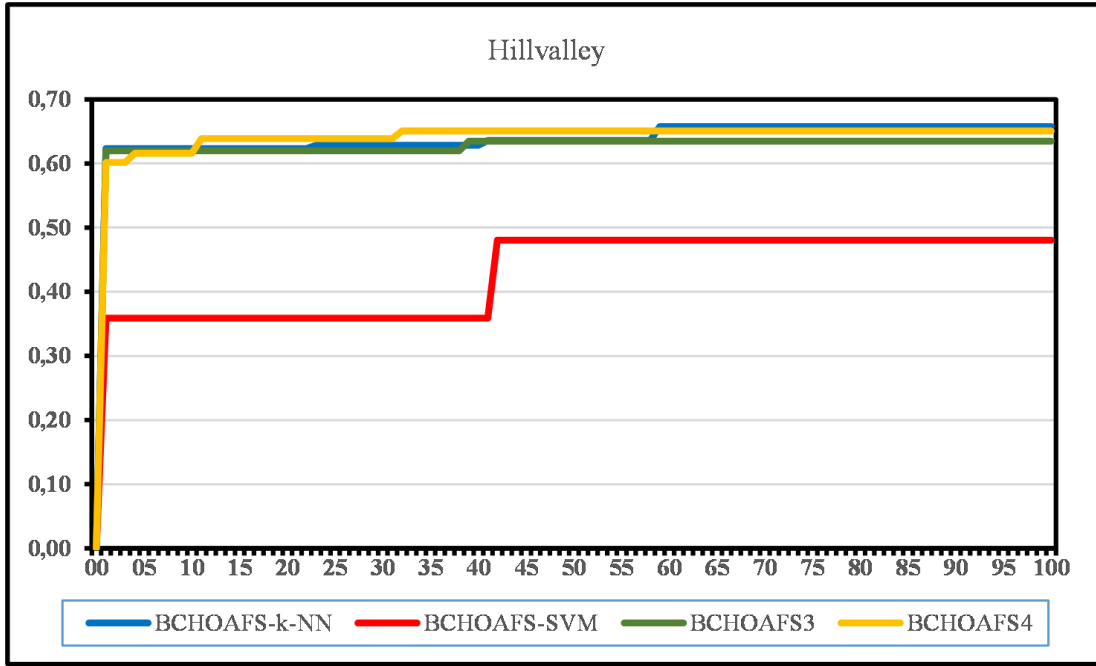
Şekil 4.3. Sonar veri seti için yakınsama grafiği.



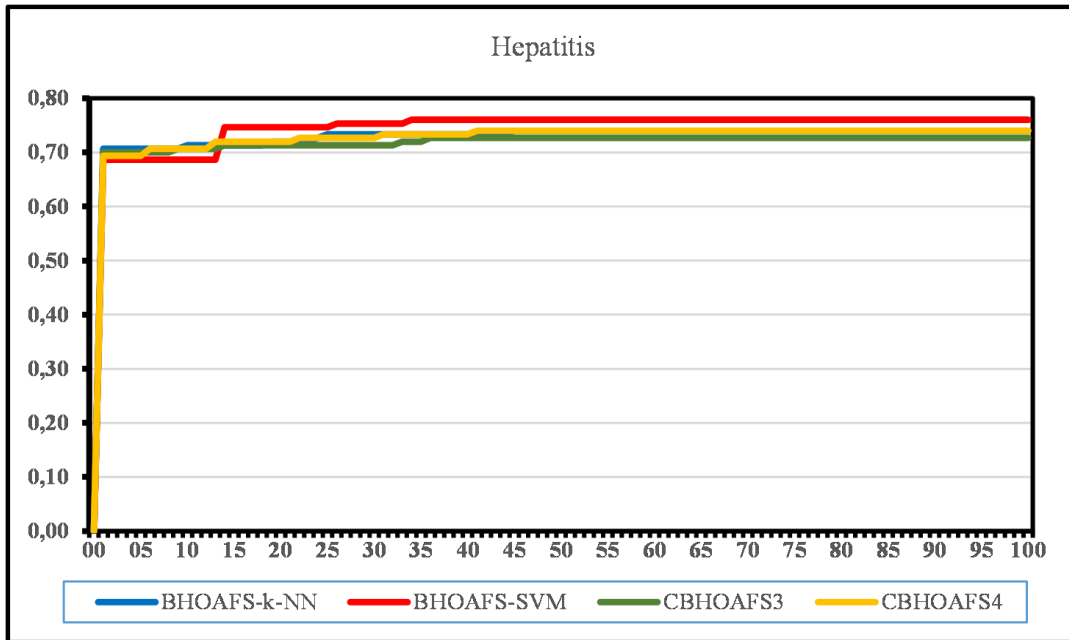
Şekil 4.4. BreastEW veri seti için yakınsama grafiği.



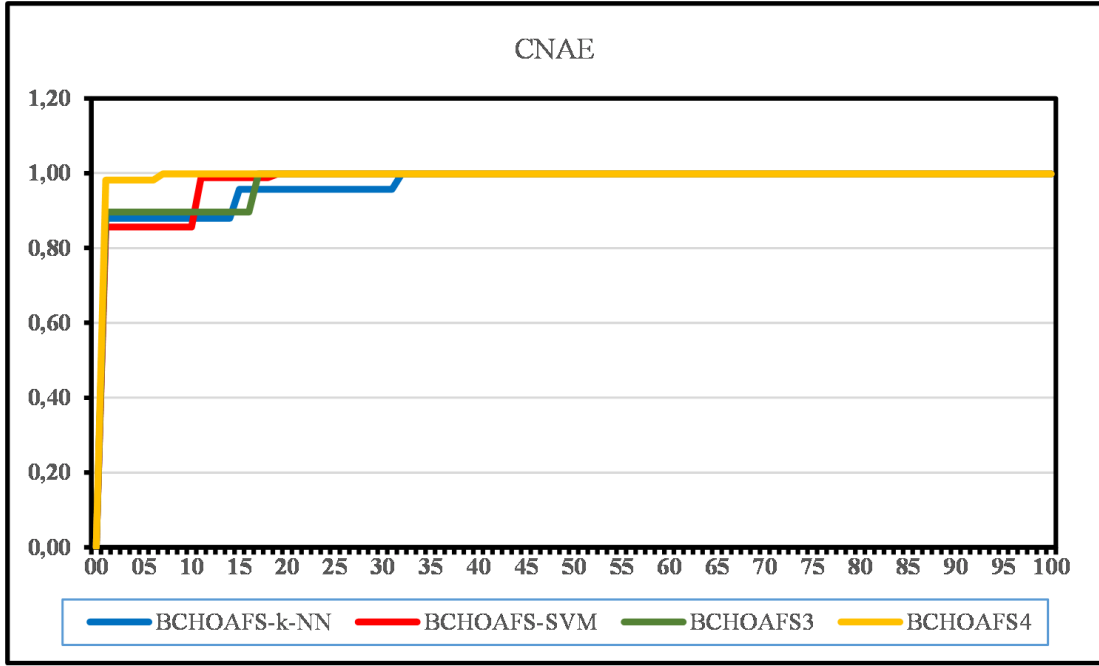
Şekil 4.5. Arrhythmia veri seti için yakınsama grafiği.



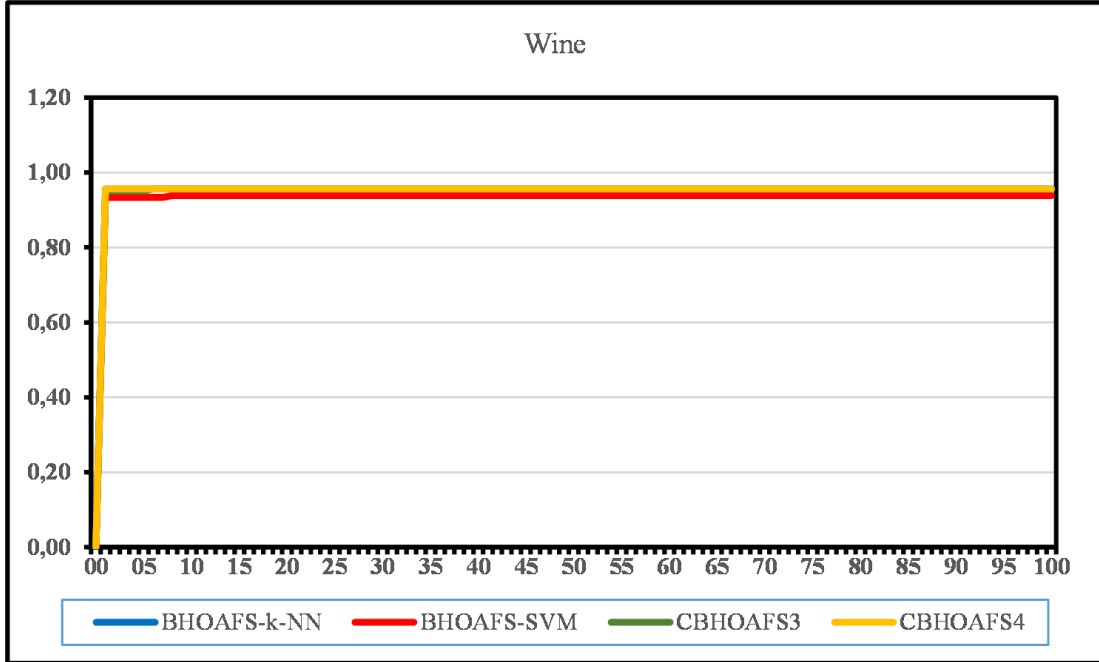
Şekil 4.6. Hillvalley veri seti için yakınsama grafiği.



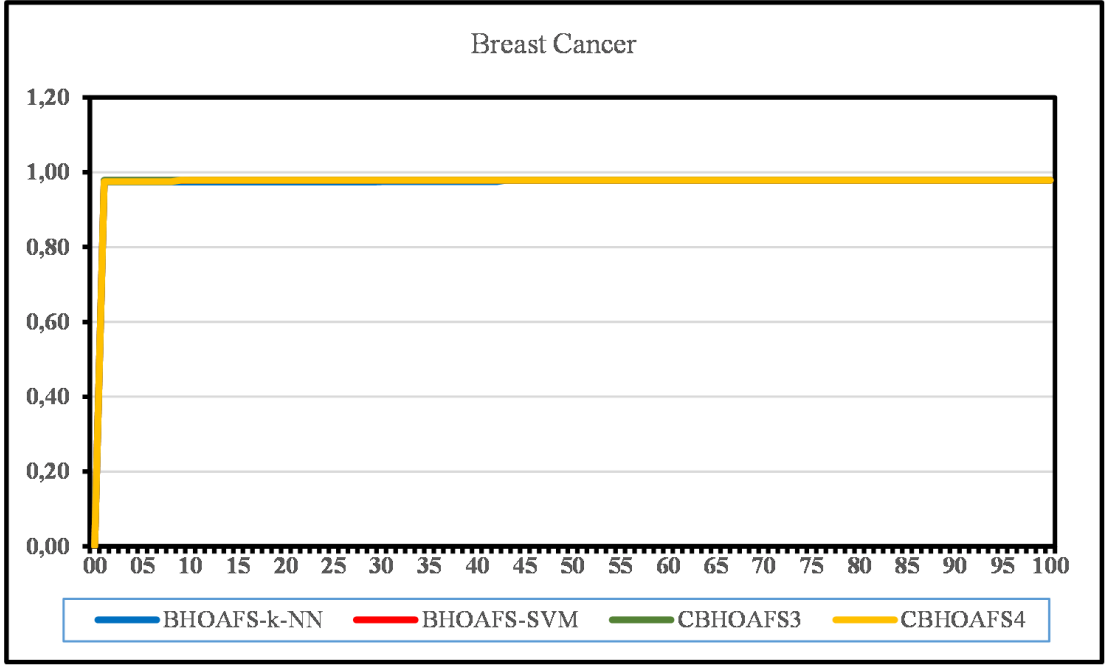
Şekil 4.7. Hepatitis veri seti için yakınsama grafiği.



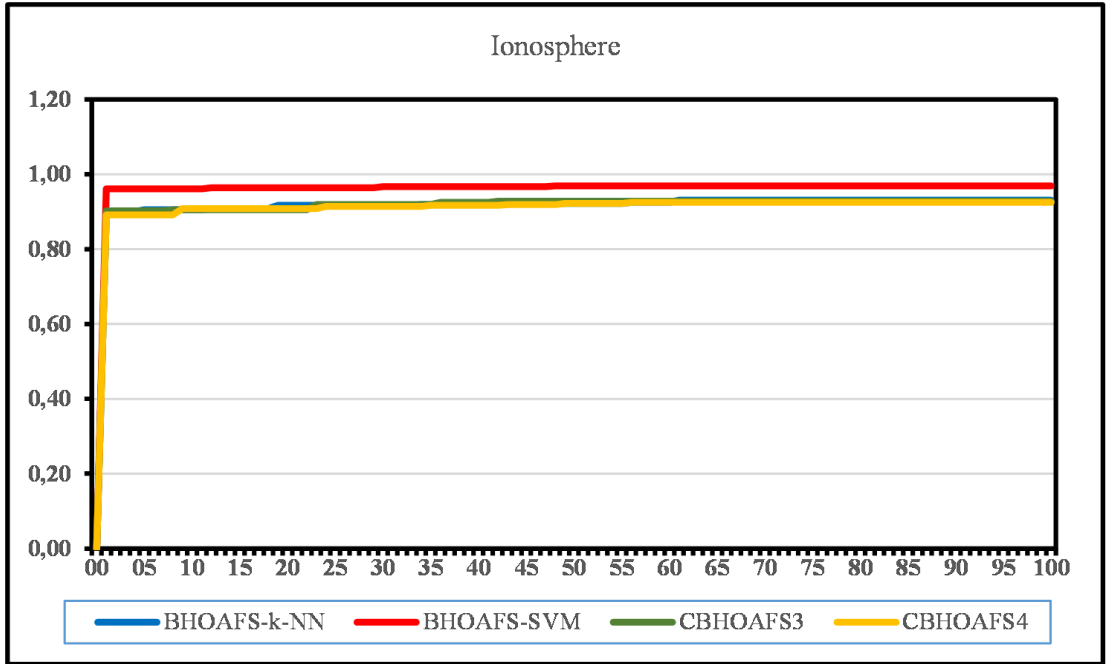
Şekil 4.8. CNAE veri seti için yakınsama grafiği.



Şekil 4.9. Wine veri seti için yakınsama grafiği.

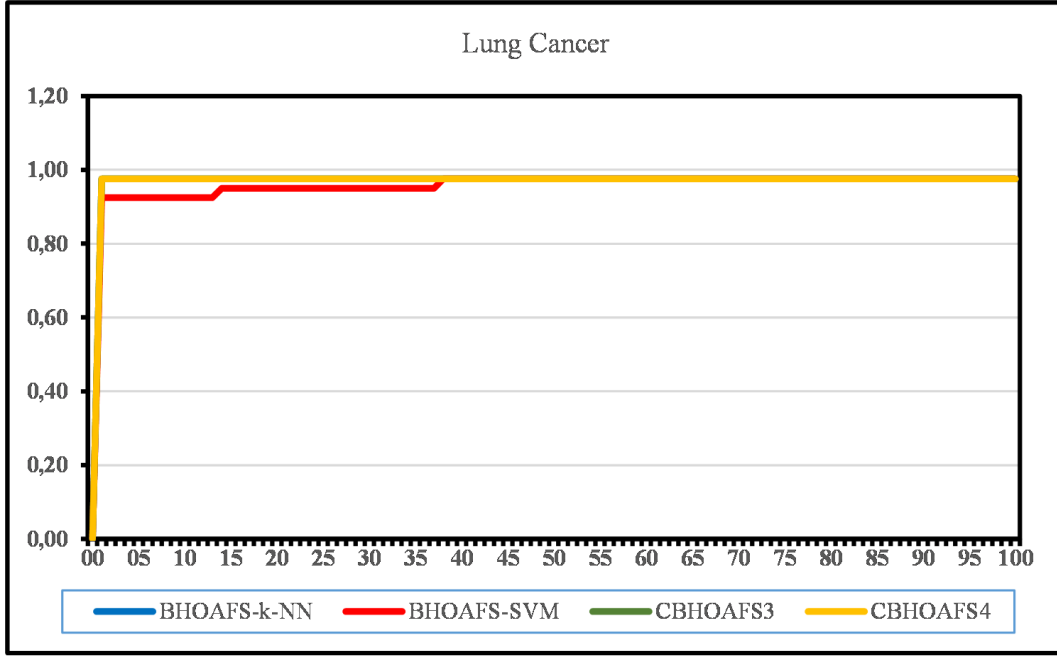


Şekil 4.10. Breast Cancer veri seti için yakınsama grafiği.

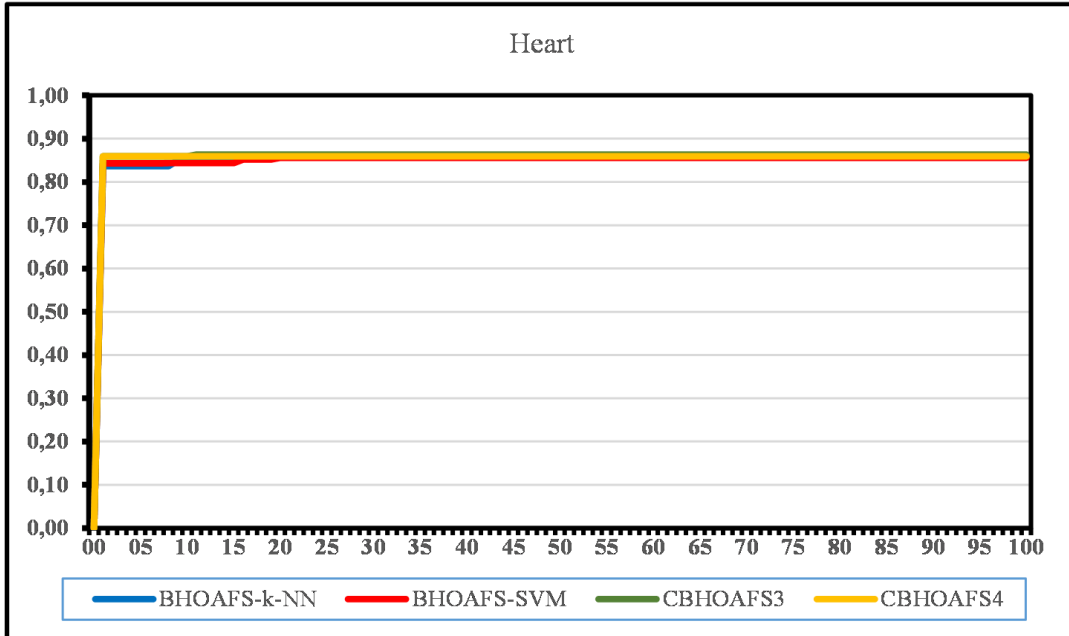


Şekil 4.11. Ionosphere veri seti için yakınsama grafiği.

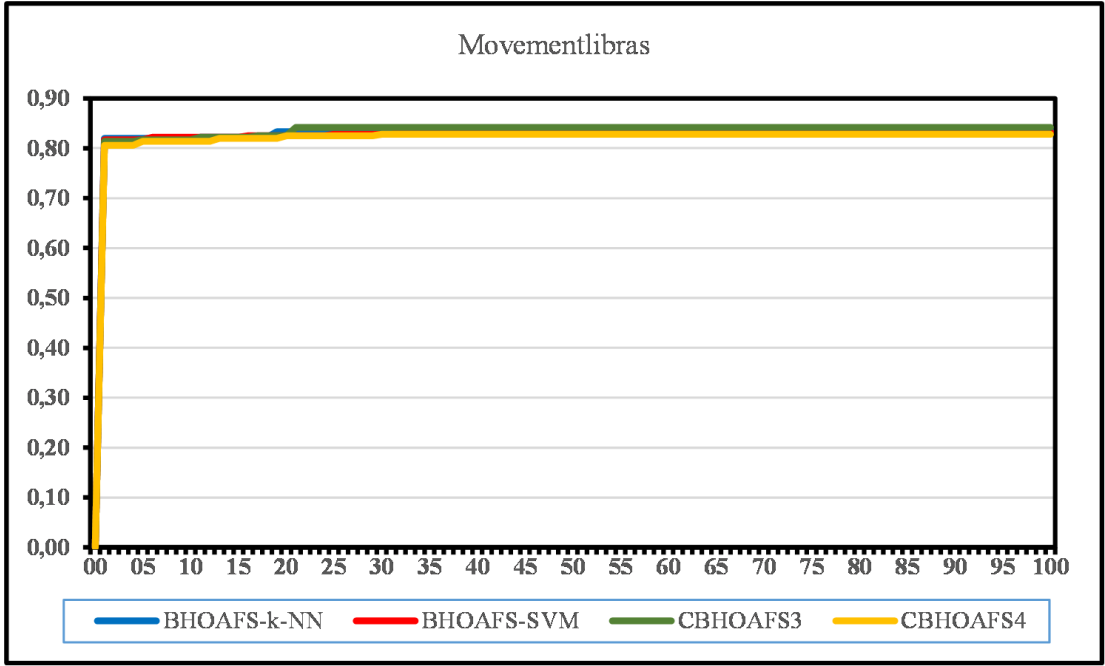




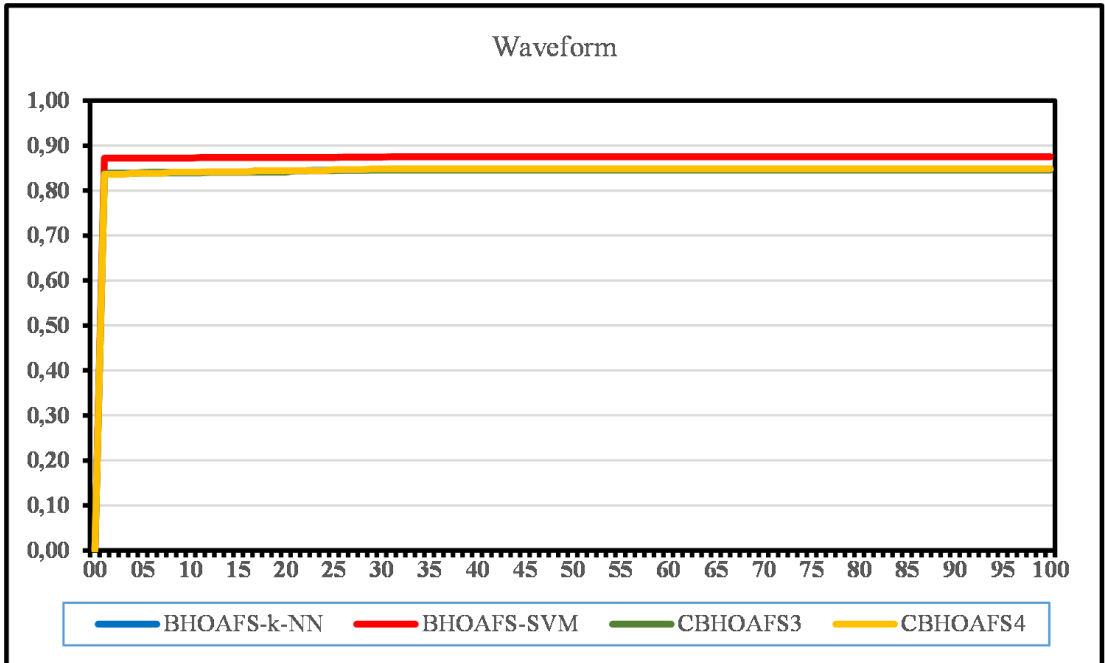
Şekil 4.12. Lung Cancer veri seti için yakınsama grafiği.



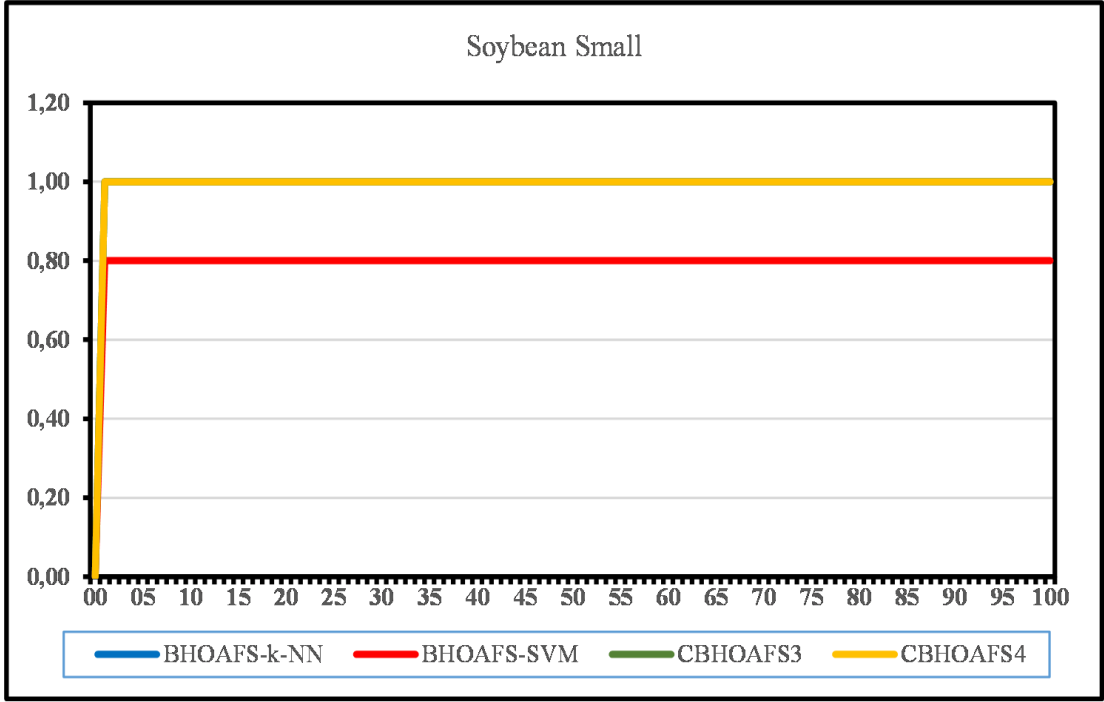
Şekil 4.13. Heart veri seti için yakınsama grafiği.



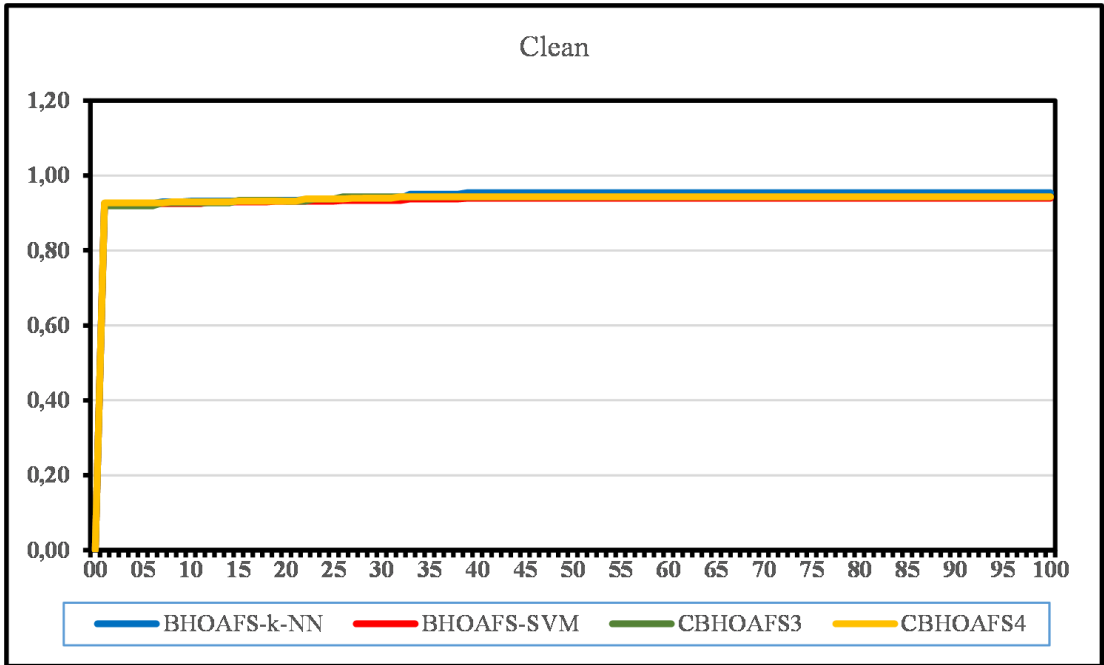
Şekil 4.14. Movementlibras veri seti için yakınsama grafiği.



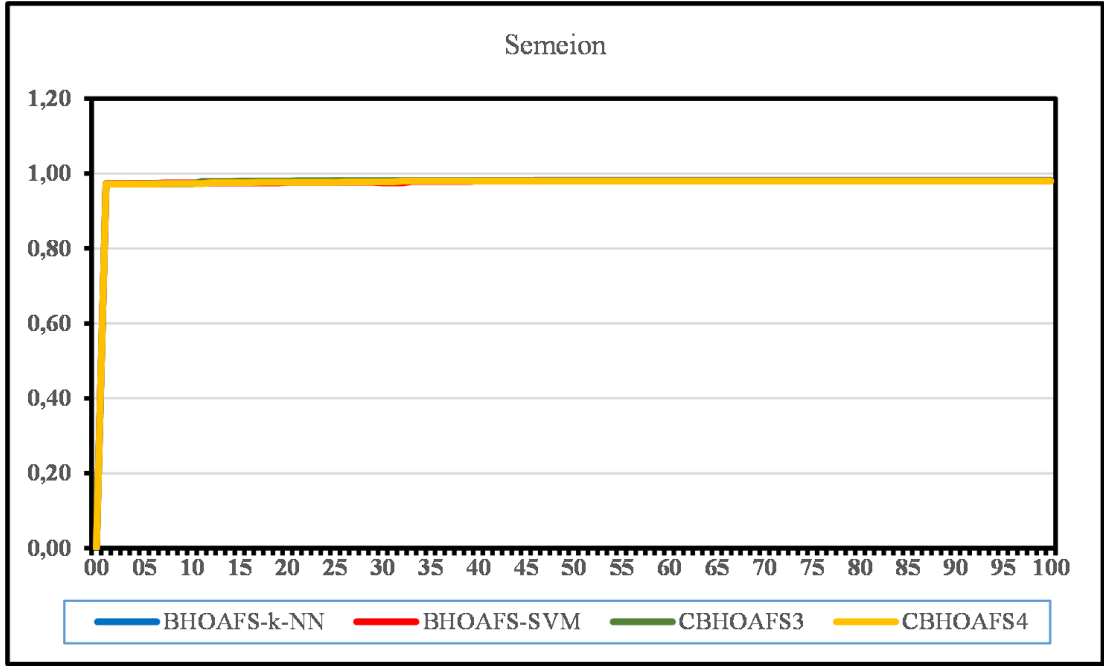
Şekil 4.15. Waveform veri seti için yakınsama grafiği.



Şekil 4.16. Soybean Small veri seti için yakınsama grafiği.



Şekil 4.17. Clean veri seti için yakınsama grafiği.



Şekil 4.18. Semeion veri seti için yakınsama grafiği.

## 4.5. İstatistiksel Test Sonuçları

### 4.5.1. Friedman test sonuçları

Friedman Testi için Hipotezler aşağıdaki gibi belirlenmiştir.

H0: Karşılaştırılan algoritmalar birbirinden farklı değildir.

H1: Karşılaştırılan algoritmalar birbirinden farklıdır.

Hipotezin reddedilip reddedilmediğini belirleyen istatistiksel test olasılık değeri  $\alpha$ 'dır. Testin tolerans eşiği  $\alpha = 0,05$ 'tir.  $df=10$  ve  $\alpha=0.05$  için beklenen  $X^2$  değeri 18.31'dir (Sheskin, 2020).  $X^2$ 'nin hesaplanan değeri beklenen değerden yüksekse ve p değeri  $\alpha = 0,5$ 'ten küçükse sıfır hipotezi reddedilebilmektedir. Tablo 4.10'da  $X^2$ 'nin hesaplanan değerinin 55.611 olduğu ve  $p = 2.4274E - 8$  olduğunu görülmektedir. Hesaplanan  $X^2$  beklenen değerden yüksektir ve p,  $\alpha$ 'dan küçüktür. Bu sonuç, sıfır hipotezinin reddedilmesi gerektiği anlamına gelir. Bu durumda önerilen algoritmanın karşılaştırılan algoritmalarından farklı olduğunu sonucu ortaya çıkmaktadır.

**Tablo 4.10.** Friedman test sonuçları.

Benchmark	Sonuç
N	17
X <sup>2</sup>	55.611
df	10
p	2.4274E - 8

Tablo 4.11 'de önerilen algoritmalara ait Friedman istatistiksel test sonuçları ve Tablo 4.12'de Friedman ortalama rank değeri kapsamıca görülmektedir.

**Tablo 4.11.** Önerilen algoritmalara ait friedman istatistiksel test sonuçları.

Algoritmalar	N	Ortalama	Std. Sapma	Minimum	Maximum	Persentil 25th
BHOAFS-k-NN	17	0.884628	0.111062	0.65574	1	0.7957
BHOAFS-SVM	17	0.859509	0.14165	0.48033	0.99815	0.78
GA	17	0.780447	0.142688	0.482	0.957	0.70325
PSO	17	0.7718	0.131694	0.5507	0.951	0.65805
ALO	17	0.7723	0.140605	0.5056	0.95	0.67055
GWO	17	0.771888	0.143456	0.5014	0.953	0.6519
SSA	17	0.8682	0.105085	0.6658	1	0.7954
BGA	17	0.780447	0.142688	0.482	0.957	0.70325
BPSO	17	0.7718	0.131694	0.5507	0.951	0.65805
BALO	17	0.7723	0.140605	0.5056	0.95	0.67055
BGWO	17	0.771888	0.143456	0.5014	0.953	0.6519
BSSA	17	0.815259	0.209434	0.1	0.975	0.75505
BCHOAFS1	17	0.884472	0.112254	0.64098	1	0.787465
BCHOAFS2	17	0.882111	0.114044	0.63607	1	0.79549
BCHOAFS3	17	0.883264	0.114165	0.63443	1	0.79966
BCHOAFS4	17	0.882069	0.110553	0.65082	1	0.791535
BCHOAFS5	17	0.838214	0.224003	0.08488	1	0.74216

**Tablo 4.12.** Önerilen algoritmalara ait friedman ortalama rank değeri.

Ranks	Ortalama Rank
BHOAFS-k-NN	13.21
BHOAFS-SVM	10.74
GA	6.85
PSO	5.32
ALO	5.44
GWO	5.26
SSA	11.24
BGA	6.85
BPSO	5.32
BALO	5.44
BGWO	5.26
BSSA	10.47
BCHOAFS1	13.03
BCHOAFS2	12.15
BCHOAFS3	12.82
BCHOAFS4	11.97
BCHOAFS5	11.62

#### 4.5.2. Wilcoxon test sonuçları

Wilcoxon testi için algoritmalarındaki farklılıklar p değerleri ile tanımlanır. p-değeri  $\alpha = 0.05$  anlamlılık değerinden küçükse, algoritmaların sonuçlarının karşılaştırılan algoritmalarından farklı olduğunu söylenebilmektedir. P değeri  $\alpha = 0.05$  anlamlılık değerinden daha yüksekse, bu durum, algoritmanın kıyaslanan algoritmalar ile benzer sonuçlara sahip olduğu anlamına gelmektedir. Tablo 4.13'te algoritmaya ait Wilcoxon istatistiksel test sonuçları değerlendirildiğinde, p değerlerinin  $\alpha = 0.05$  anlamlılık değerinden daha küçük olduğu görülmektedir. Bu değer önerilen algoritmanın karşılaştırılan algoritmaların çoğundan farklı sonuçlara sahip olduğu manasına gelmektedir. BSSA algoritması sonuçları incelendiğinde önerilen algoritma ile benzer sonuçlara sahip olduğu söylenebilir ancak Tablo 4.6 ve Tablo 4.7'deki doğruluk sonuçlarına bakıldığında, önerilen algoritmanın daha iyi sonuçlar verdiğini görülmektedir. Tablo 4.14 'de önerilen algoritmalara ait Wilcoxon istatistiksel test sonuçları kapsamlıca görülmektedir.



**Tablo 4.13.** Önerilen algoritmaların wilcoxon istatistiksel testine ait p değeri (anlamlılık düzeyi 0.05).

Algorithm	BHOAFS - k-NN	BCHOAF S1	BCHOAF S2	BCHOAF S3	BCHOAF S4	BCHOAF S5	BGA	BPSO	BALO	BGWO	BSSA
BHOAFS-k- NN	1	0.646	0.328	0.594	0.050	0.388	<b>0.022</b>	<b>0.006</b>	<b>0.010</b>	<b>0.009</b>	0.177
BCHOAFS1		1	0.176	0.484	0.433	0.674	<b>0.017</b>	<b>0.004</b>	<b>0.010</b>	<b>0.007</b>	0.149
BCHOAFS2			1	0.161	0.790	0.612	<b>0.025</b>	<b>0.006</b>	<b>0.011</b>	<b>0.010</b>	0.163
BCHOAFS3				1	0.328	0.398	<b>0.019</b>	<b>0.006</b>	<b>0.010</b>	<b>0.010</b>	0.163
BCHOAFS4					1	0.534	<b>0.022</b>	<b>0.007</b>	<b>0.015</b>	<b>0.011</b>	0.227
BCHOAFS5						1	<b>0.022</b>	<b>0.006</b>	<b>0.011</b>	<b>0.009</b>	0.136

**Tablo 4.14.** Önerilen algoritmaya ait wilcoxon istatistiksel test sonuçları.

Algoritmalar	N	Ortalama	Std. Sapma	Minimum	Maximum	Persentil 25th
BHOAFS-k-NN	17	0.884628	0.111062	0.65574	1	0.7957
BCHOAFS1	17	0.884472	0.112254	0.64098	1	0.787465
BCHOAFS2	17	0.882111	0.114044	0.63607	1	0.79549
BCHOAFS3	17	0.883264	0.114165	0.63443	1	0.79966
BCHOAFS4	17	0.882069	0.110553	0.65082	1	0.791535
BCHOAFS5	17	0.838214	0.224003	0.08488	1	0.74216
BGA	17	0.780447	0.142688	0.482	0.957	0.70325
BPSO	17	0.7718	0.131694	0.5507	0.951	0.65805
BALO	17	0.7723	0.140605	0.5056	0.95	0.67055
BGWO	17	0.771888	0.143456	0.5014	0.953	0.6519
BSSA	17	0.815259	0.209434	0.1	0.975	0.75505

## 5. SONUÇ VE ÖNERİLER

Öznitelik seçim problemi gereksiz/verimsiz verilerin ortadan kaldırılması açısından veri madenciliği ve makine öğrenmesi alanında hala zor bir problem olarak görülmektedir. Literatürdeki çalışmalar detaylı olarak incelendiğinde öznitelik seçimi problem çözümü için daha verimli teknikler geliştirme imkanının halen mümkün olduğu görülmektedir. En iyi performansa sahip en iyi özellik alt kümesinin seçimi çeşitli metodlar kullanılarak büyük ölçüde geliştirilmiş olsa da, yeni yayınlanan optimizasyon teknikleri ile yapılan çalışmalar hala iyileştirmeye açıktır.

Bu tez çalışmasında HOA'nın ikili versiyonu olan ikili at sürüsü optimizasyon algoritması, uygun özellik alt kümesini seçmek için öznitelik seçme problemine uygulanmıştır. Büyük boyutlu veri için özel olarak geliştirilen At Sürüsü Optimizasyon Algoritmasının önerilen versiyonu (BHOAFS) ile son yıllarda oldukça popüler olan keşif ve sömürü süreci üzerine yapılan çalışmalara katkı sağlanmıştır. k-NN ve SVM kullanılarak geliştirilen algoritma sınıflandırma doğruluğu üzerinden test edilmiştir. Deneysel sonuçlar, BHOAFS-k-NN'nin BHOAFS-SVM'den daha iyi sonuçlar ürettiğini göstermiştir. Önerilen algoritma, kaotik haritalar eklenmeden önce BHOAFS olarak adlandırılmıştır. Ardından önerilen algoritma BHOAFS-k-NN'ye beş tek boyutlu kaotik harita (Logistic, Tent, Piecewise, Singer ve Sinusoidal) eklenerek İkili Kaotik At Sürü Optimizasyon Algoritması (BCHOAFS) önerilmiştir. Sınıflandırma doğruluğunu en üst düzeye çıkarmak, veri setine ait özellik sayısını en aza indirerek en uygun öznitelikleri seçmek ve algoritmanın kararlılığını artırmak için kaotik haritalar ekleyerek çok amaçlı kaotik arama tabanlı BCHOAFS önerilmiştir. Önerilen BCHOAFS yöntemlerinin performansı, UCI deposundan alınan 18 farklı veri seti üzerinde çalıştırılarak, literatürce iyi bilinen algoritmalar ile değerlendirilmiştir. Deneysel sonuçlar önerilen BCHOAFS'nin, sınıflandırma doğruluğu, seçilen öznitelik sayısı açısından değerlendirildiğinde kıyaslanan diğer algoritmalarından daha iyi sonuçlar elde ettiğini göstermektedir.

Tüm bu sonuçlara rağmen NFL Teoremi tüm olası optimizasyon problemlerini çözebilecek hiçbir optimizasyon algoritmasının bulunmadığını iddia eder(Wolpert &

Macready, 1997). Bu teorem her optimizasyon yönteminin her problemi en iyi şekilde çözemeyeceği ve her yöntemin her zaman daha başarılı olabileceği bir başka problem bulunabileceği esasına dayanır. Tüm bu varsayımlara dayanarak hali hazırda önerilen optimizasyon algoritmalarını geliştirmek, iyileştirmek veya özelleştirmek için hala geçerli sebepler bulunmaktadır.

Gelecek çalışmalarda önerilen yöntemde SMF işlevi ile arama uzayının merkezine doğru yönlendirilen at sürüsü hareketi, hareket duyarlılığı hesaplanarak keşif kabiliyeti açısından farklı bir çözüm ile geliştirilebilir. HOA başka algoritmalar ile birleştirilerek çok amaçlı sürümün iyileştirilmesi sağlanabilir. Arama uzayındaki her atın konumuna ait özelleştirilen çözümler çeşitli istatistikî yöntemler ile desteklenerek geliştirilebilir. Sürekli uzaydan ayrık uzaya geçişte kullanılan U-şekilli transfer fonksiyonu başka transfer fonksiyonları ile değiştirilerek denenebilir. Öznitelik seçim performans ölçümü için kullanılan sınıflandırıcılar (k-NN ve SVM dışında) başka sınıflandırma algoritmaları ile değiştirilerek doğruluğu karşılaştırılabilir.

## KAYNAKLAR

- Abdel-Raouf, O., Abdel-Baset, M., & El-Henawy, I. (2014). An improved chaotic bat algorithm for solving integer programming problems. *International Journal of Modern Education and Computer Science*, 6(8), 18.
- Abualigah, L., & Diabat, A. (2022). Chaotic binary Group Search Optimizer for feature selection. *Expert Systems with Applications*, 192, 116368. <https://doi.org/10.1016/j.eswa.2021.116368>
- Agustian, F., & Lubis, M. D. I. (2020). Particle Swarm Optimization Feature Selection for Breast Cancer Prediction. 2020 8th International Conference on Cyber and IT Service Management (CITSM), 1-6.
- Akila, S., & Allin Christe, S. (2022). A wrapper based binary bat algorithm with greedy crossover for attribute selection. *Expert Systems with Applications*, 187, 115828. <https://doi.org/10.1016/j.eswa.2021.115828>
- Alatas, B., Akin, E., & Ozer, A. B. (2009). Chaos embedded particle swarm optimization algorithms. *Chaos, Solitons & Fractals*, 40(4), 1715-1734.
- Albashish, D., Hammouri, A. I., Braik, M., Atwan, J., & Sahran, S. (2021). Binary biogeography-based optimization based SVM-RFE for feature selection. *Applied Soft Computing*, 101, 107026. <https://doi.org/10.1016/j.asoc.2020.107026>
- Ali, W., & Ahmed, A. A. (2019). Hybrid intelligent phishing website prediction using deep neural networks with genetic algorithm-based feature selection and weighting. *IET Information Security*, 13(6), 659-669. <https://doi.org/10.1049/iet-ifs.2019.0006>
- Amini, F., & Hu, G. (2021). A two-layer feature selection method using Genetic Algorithm and Elastic Net. *Expert Systems with Applications*, 166, 114072. <https://doi.org/10.1016/j.eswa.2020.114072>
- Amiri, A. M., & Armano, G. (2013). Segmentation and feature extraction of heart murmurs in newborns. *Journal of Life Sciences and Technologies Vol*, 1(2).
- Anaraki, J. R., Samet, S., Eftekhari, M., & Ahn, C. W. (2018). A Fuzzy-Rough based Binary Shuffled Frog Leaping Algorithm for Feature Selection. arXiv preprint arXiv:1808.00068.
- Arora, S., & Anand, P. (2019). Binary butterfly optimization approaches for feature selection. *Expert Systems with Applications*, 116, 147-160. <https://doi.org/10.1016/j.eswa.2018.08.051>

- Awadallah, M. A., Hammouri, A. I., Al-Betar, M. A., Braik, M. S., & Elaziz, M. A. (2022). Binary Horse herd optimization algorithm with crossover operators for feature selection. *Computers in Biology and Medicine*, 141, 105152. <https://doi.org/10.1016/j.combiomed.2021.105152>
- Azar, A. T., Banu, N., & Koubaa, A. (2020). Rough Set Based Ant-Lion Optimizer for Feature Selection. 2020 6th Conference on Data Science and Machine Learning Applications (CDMA), 81-86.
- Baş, E., & Ülker, E. (2020). An efficient binary social spider algorithm for feature selection problem. *Expert Systems with Applications*, 146, 113185. <https://doi.org/10.1016/j.eswa.2020.113185>
- Batista, G., & Silva, D. F. (2009). How k-nearest neighbor parameters affect its performance. *Argentine symposium on artificial intelligence*, 1-12.
- Bhatia, N. (2010). Survey of nearest neighbor techniques. *arXiv preprint arXiv:1007.0085*.
- Brezočnik, L., Fister, I., & Podgorelec, V. (2018). Swarm Intelligence Algorithms for Feature Selection: A Review. *Applied Sciences*, 8(9), 1521. <https://doi.org/10.3390/app8091521>
- Brownlee, J. (2020). Data preparation for machine learning: Data cleaning, feature selection, and data transforms in Python. *Machine Learning Mastery*.
- Canayaz, M. (2021). MH-COVIDNet: Diagnosis of COVID-19 using deep neural networks and meta-heuristic-based feature selection on X-ray images. *Biomedical Signal Processing and Control*, 64, 102257. <https://doi.org/10.1016/j.bspc.2020.102257>
- Chen, K., & Kanoh, H. (2016). A Discrete Artificial Bee Colony Algorithm Based on Similarity for Graph Coloring Problems. *International Conference on Theory and Practice of Natural Computing*, 73-84.
- Chen, Y., Zhu, Q., & Xu, H. (2015). Finding rough set reducts with fish swarm algorithm. *Knowledge-Based Systems*, 81, 22-29.
- Conover, W. J. (1999). *Practical nonparametric statistics* (C. 350). John Wiley & Sons.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20, 273-297.
- Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1), 21-27.
- Dasgupta, A., Banerjee, A., Dastidar, A. G., Barman, A., & Chakraborty, S. (2019). A Study and Analysis of a Feature Subset Selection Technique Using Penguin Search Optimization Algorithm. *Içinde Q. A. Memon & S. A. Khoja (Ed.), Data Science* (1. bs, ss. 51-68). CRC Press. <https://doi.org/10.1201/9780429263798-3>

- Demir, F. B., Tuncer, T., & Kocamaz, A. F. (2020). A chaotic optimization method based on logistic-sine map for numerical function optimization. *Neural Computing and Applications*, 32, 14227-14239.
- Department of Operations Research, Faculty of Computers and Information, Menoufia University, Menoufia, Shebin-El-coma, Egypt., Abdel Raouf, O., Abdel Baset, M., & El henawy, I. (2014). An Improved Chaotic Bat Algorithm for Solving Integer Programming Problems. *International Journal of Modern Education and Computer Science*, 6(8), 18-24. <https://doi.org/10.5815/ijmecs.2014.08.03>
- ERDOĞMUŞ, P. (2016). Doğadan esinlenen optimizasyon algoritmaları ve optimizasyon algoritmalarının optimizasyonu. *Düzce üniversitesi bilim ve teknoloji dergisi*, 4(1), 293-304.
- Ewees, A. A., El Aziz, M. A., & Hassanien, A. E. (2019). Chaotic multi-verse optimizer-based feature selection. *Neural Computing and Applications*, 31(4), 991-1006. <https://doi.org/10.1007/s00521-017-3131-4>
- Gandomi, A. H., Yang, X.-S., Talatahari, S., & Alavi, A. H. (2013). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, 18(1), 89-98.
- Gärtner, T., Lloyd, J. W., & Flach, P. A. (2004). Kernels and distances for structured data. *Machine Learning*, 57, 205-232.
- Ghosh, K. K., Ahmed, S., Singh, P. K., Geem, Z. W., & Sarkar, R. (2020). Improved Binary Sailfish Optimizer Based on Adaptive  $\beta$  -Hill Climbing for Feature Selection. *IEEE Access*, 8, 83548-83560. <https://doi.org/10.1109/ACCESS.2020.2991543>
- Got, A., Moussaoui, A., & Zouache, D. (2021). Hybrid filter-wrapper feature selection using whale optimization algorithm: A multi-objective approach. *Expert Systems with Applications*, 183, 115312. <https://doi.org/10.1016/j.eswa.2021.115312>
- Guntsch, M., & Middendorf, M. (2002). A population based approach for ACO. *Applications of Evolutionary Computing: EvoWorkshops 2002: EvoCOP, EvoIASP, EvoSTIM/EvoPLAN Kinsale, Ireland, April 3–4, 2002 Proceedings*, 72-81.
- Han, J., & Kamber, M. (2006). Classification and prediction. *Data mining: Concepts and techniques*, 2006, 347-350.
- Hancer, E., Xue, B., Zhang, M., Karaboga, D., & Akay, B. (2018). Pareto front feature selection based on artificial bee colony optimization. *Information Sciences*, 422, 462-479. <https://doi.org/10.1016/j.ins.2017.09.028>
- Hichem, H., Rafik, M., & Mesaaoud, M. T. (2018). PSO with crossover operator applied to feature selection problem in classification. *Informatika*, 42(2).

- Hu, J., Gui, W., Heidari, A. A., Cai, Z., Liang, G., Chen, H., & Pan, Z. (2021). Dispersed foraging slime mould algorithm: Continuous and binary variants for global optimization and wrapper-based feature selection. *Knowledge-Based Systems*, 107761. <https://doi.org/10.1016/j.knosys.2021.107761>
- Hu, X.-M., He, F.-L., Chen, W.-N., & Zhang, J. (2017). Cooperation coevolution with fast interdependency identification for large scale optimization. *Information Sciences*, 381, 142-160.
- Hu, Y., Zhu, C., & Wang, Z. (2014). An improved piecewise linear chaotic map based image encryption algorithm. *The Scientific World Journal*, 2014.
- Jia, D., Duan, X., & Khan, M. K. (2014). Binary Artificial Bee Colony optimization using bitwise operation. *Computers & Industrial Engineering*, 76, 360-365.
- Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, 4, 580-585.
- Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. 1997 IEEE International conference on systems, man, and cybernetics. *Computational cybernetics and simulation*, 5, 4104-4108.
- Khare, N., Devan, P., Chowdhary, C., Bhattacharya, S., Singh, G., Singh, S., & Yoon, B. (2020). SMO-DNN: Spider Monkey Optimization and Deep Neural Network Hybrid Classifier Model for Intrusion Detection. *Electronics*, 9(4), 692. <https://doi.org/10.3390/electronics9040692>
- Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2), 273-324. [https://doi.org/10.1016/S0004-3702\(97\)00043-X](https://doi.org/10.1016/S0004-3702(97)00043-X)
- Kushwaha, N., & Pant, M. (2018). Link based BPSO for feature selection in big data text clustering. *Future Generation Computer Systems*, 82, 190-199.
- Li, X. (2002). An optimizing method based on autonomous animats: Fish-swarm algorithm. *Systems Engineering-Theory & Practice*, 22(11), 32-38.
- Liu, H., & Zhang, S. (2012). Noisy data elimination using mutual k-nearest neighbor for classification mining. *Journal of Systems and Software*, 85(5), 1067-1074.
- Liu, Z., Chang, B., & Cheng, F. (2021). An interactive filter-wrapper multi-objective evolutionary algorithm for feature selection. *Swarm and Evolutionary Computation*, 65, 100925. <https://doi.org/10.1016/j.swevo.2021.100925>
- Mafarja, M. M., Eleyan, D., Jaber, I., Hammouri, A., & Mirjalili, S. (2017a). Binary dragonfly algorithm for feature selection. 2017 International conference on new trends in computing sciences (ICTCS), 12-17.
- Mafarja, M. M., Eleyan, D., Jaber, I., Hammouri, A., & Mirjalili, S. (2017b). Binary dragonfly algorithm for feature selection. 2017 International conference on new trends in computing sciences (ICTCS), 12-17.



- Manoj, R. J., Praveena, M. A., & Vijayakumar, K. (2019). An ACO–ANN based feature selection algorithm for big data. *Cluster Computing*, 22(2), 3953-3960.
- Mao, C., Hu, B., Wang, M., & Moore, P. (2015). Learning from neighborhood for classification with local distribution characteristics. 2015 International Joint Conference on Neural Networks (IJCNN), 1-8.
- Maria, M., & Yassine, C. (2020). Machine learning based approaches for modeling the output power of photovoltaic array in real outdoor conditions. *Electronics*, 9(2), 315.
- McCrum-Gardner, E. (2008). Which is the correct statistical test to use? *British Journal of Oral and Maxillofacial Surgery*, 46(1), 38-41.
- MiarNaeimi, F., Azizyan, G., & Rashki, M. (2021a). Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems. *Knowledge-Based Systems*, 213, 106711. <https://doi.org/10.1016/j.knosys.2020.106711>
- MiarNaeimi, F., Azizyan, G., & Rashki, M. (2021b). Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems. *Knowledge-Based Systems*, 213, 106711.
- Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-based systems*, 89, 228-249.
- Mirjalili, S., Zhang, H., Mirjalili, S., Chalup, S., & Noman, N. (2020). A novel U-shaped transfer function for binary particle swarm optimisation. *Soft Computing for Problem Solving 2019: Proceedings of SocProS 2019, Volume 1*, 241-259.
- Mitchell, T. M. (2007). *Machine learning (C. 1)*. McGraw-hill New York.
- Muhammad, Z. M. Z., & Özkaynak, F. (2020). An image encryption algorithm based on chaotic selection of robust cryptographic primitives. *IEEE Access*, 8, 56581-56589.
- Nguyen, B. H., Xue, B., & Zhang, M. (2020). A survey on swarm intelligence approaches to feature selection in data mining. *Swarm and Evolutionary Computation*, 54, 100663. <https://doi.org/10.1016/j.swevo.2020.100663>
- Nouri-Moghaddam, B., Ghazanfari, M., & Fathian, M. (2021). A novel multi-objective forest optimization algorithm for wrapper feature selection. *Expert Systems with Applications*, 175, 114737. <https://doi.org/10.1016/j.eswa.2021.114737>
- Osuna, E. E. (1998). *Support vector machines: Training and applications*. Massachusetts Institute of Technology.
- Ouadfel, S., & Abd Elaziz, M. (2020). Enhanced crow search algorithm for feature selection. *Expert Systems with Applications*, 159, 113572. <https://doi.org/10.1016/j.eswa.2020.113572>

- Ouadfel, S., & Abd Elaziz, M. (2022). Efficient high-dimension feature selection based on enhanced equilibrium optimizer. *Expert Systems with Applications*, 187, 115882. <https://doi.org/10.1016/j.eswa.2021.115882>
- Piri, J., & Mohapatra, P. (2021). An analytical study of modified multi-objective Harris Hawk Optimizer towards medical data feature selection. *Computers in Biology and Medicine*, 135, 104558. <https://doi.org/10.1016/j.combiomed.2021.104558>
- Qasim, O. S., Al-Thanoon, N. A., & Algamal, Z. Y. (2020). Feature selection based on chaotic binary black hole algorithm for data classification. *Chemometrics and Intelligent Laboratory Systems*, 204, 104104. <https://doi.org/10.1016/j.chemolab.2020.104104>
- Qiu, H., & Liu, Y. (2016). Novel Heuristic Algorithm for Large-scale Complex Optimization. *ICCS*, 744-751.
- Rajabhushanam, C. (2020). A SURVEY ON FEATURE SELECTION TO PERFORM CLASSIFICATION USING META HEURISTIC ALGORITHMS IN DATA MINING DOMAIN. *Journal of Critical Reviews*, 7(10), 2031-2042.
- Rodrigues, D., de Albuquerque, V. H. C., & Papa, J. P. (2020). A multi-objective artificial butterfly optimization approach for feature selection. *Applied Soft Computing*, 94, 106442. <https://doi.org/10.1016/j.asoc.2020.106442>
- S., P., Al-Turjman, F., & Stephan, T. (2021). An automated breast cancer diagnosis using feature selection and parameter optimization in ANN. *Computers & Electrical Engineering*, 90, 106958. <https://doi.org/10.1016/j.compeleceng.2020.106958>
- Sahran, S., Albashish, D., Abdullah, A., Shukor, N. A., & Pauzi, S. H. M. (2018). Absolute cosine-based SVM-RFE feature selection method for prostate histopathological grading. *Artificial Intelligence in Medicine*, 87, 78-90. <https://doi.org/10.1016/j.artmed.2018.04.002>
- Sánchez-Marroño, N., Alonso-Betanzos, A., & Tombilla-Sanromán, M. (2007a). Filter methods for feature selection—a comparative study. *International Conference on Intelligent Data Engineering and Automated Learning*, 178-187.
- Sánchez-Marroño, N., Alonso-Betanzos, A., & Tombilla-Sanromán, M. (2007b). Filter methods for feature selection—a comparative study. *International Conference on Intelligent Data Engineering and Automated Learning*, 178-187.
- Sarafrazi, S., & Nezamabadi-pour, H. (2013). Facing the classification of binary problems with a GSA-SVM hybrid system. *Mathematical and Computer Modelling*, 57(1), 270-278. <https://doi.org/10.1016/j.mcm.2011.06.048>
- Sayed, G. I., Hassanien, A. E., & Azar, A. T. (2019a). Feature selection via a novel chaotic crow search algorithm. *Neural Computing and Applications*, 31(1), 171-188. <https://doi.org/10.1007/s00521-017-2988-6>

- Sayed, G. I., Hassanien, A. E., & Azar, A. T. (2019b). Feature selection via a novel chaotic crow search algorithm. *Neural Computing and Applications*, 31(1), 171-188. <https://doi.org/10.1007/s00521-017-2988-6>
- Sayed, G. I., Tharwat, A., & Hassanien, A. E. (2019). Chaotic dragonfly algorithm: An improved metaheuristic algorithm for feature selection. *Applied Intelligence*, 49(1), 188-205. <https://doi.org/10.1007/s10489-018-1261-8>
- Sheskin, D. J. (2020). *Handbook of parametric and nonparametric statistical procedures*. crc Press.
- Sheth, P. D., Patil, S. T., & Dhore, M. L. (2020). Evolutionary computing for clinical dataset classification using a novel feature selection algorithm. *Journal of King Saud University - Computer and Information Sciences*, S1319157820306200. <https://doi.org/10.1016/j.jksuci.2020.12.012>
- Shi, Y. (2015). An optimization algorithm based on brainstorming process. *İçinde Emerging Research on Swarm Intelligence and Algorithm Optimization* (ss. 1-35). IGI Global.
- Shukla, A. K., Singh, P., & Vardhan, M. (2018). A hybrid gene selection method for microarray recognition. *Biocybernetics and Biomedical Engineering*, 38(4), 975-991. <https://doi.org/10.1016/j.bbe.2018.08.004>
- Silva, S. F. da, Ribeiro, M. X., Neto, J. do E. S. B., Traina-Jr, C., & Traina, A. J. M. (2011). Improving the ranking quality of medical image retrieval using a genetic feature selection method. *Decision Support Systems*, 51(4), 810-820. <https://doi.org/10.1016/j.dss.2011.01.015>
- Srinivasa Rao, P. C., Sravan Kumar, A. J., Niyaz, Q., Sidike, P., & Devabhaktuni, V. K. (2020). Binary chemical reaction optimization based feature selection techniques for machine learning classification problems. *Expert Systems with Applications*, 114169. <https://doi.org/10.1016/j.eswa.2020.114169>
- Sun, X., Liu, Y., Li, J., Zhu, J., Chen, H., & Liu, X. (2012). Feature evaluation and selection with cooperative game theory. *Pattern Recognition*, 45(8), 2992-3002. <https://doi.org/10.1016/j.patcog.2012.02.001>
- Taghian, S., & Nadimi-Shahraki, M. H. (2019). A binary metaheuristic algorithm for wrapper feature selection. *International Journal of Computer Science Engineering (IJCSE)*, 8(5), 168-172.
- Taheri, S. M., & Hesamian, G. (2013). A generalization of the Wilcoxon signed-rank test and its applications. *Statistical Papers*, 54(2), 457.
- Talatahari, S., Azar, B. F., Sheikholeslami, R., & Gandomi, A. H. (2012). Imperialist competitive algorithm combined with chaos for global optimization. *Communications in Nonlinear Science and Numerical Simulation*, 17(3), 1312-1319.
- Talbi, E.-G. (2009). *Metaheuristics: From design to implementation*. John Wiley & Sons.

- Taşcı, E., & Onan, A. (2016). K-en yakın komşu algoritması parametrelerinin sınıflandırma performansı üzerine etkisinin incelenmesi. *Akademik Bilişim*, 1(1), 4-18.
- Thuy, N. N., & Wongthanavas, S. (2020). A new approach for reduction of attributes based on stripped quotient sets. *Pattern Recognition*, 97, 106999.
- Tran, B., Xue, B., & Zhang, M. (2016). Genetic programming for feature construction and selection in classification on high-dimensional data. *Memetic Computing*, 8(1), 3-15.
- Tubishat, M., Idris, N., Shuib, L., Abushariah, M. A. M., & Mirjalili, S. (2020). Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Systems with Applications*, 145, 113122. <https://doi.org/10.1016/j.eswa.2019.113122>
- Tubishat, M., Ja'afar, S., Alswaiti, M., Mirjalili, S., Idris, N., Ismail, M. A., & Omar, M. S. (2021). Dynamic Salp swarm algorithm for feature selection. *Expert Systems with Applications*, 164, 113873. <https://doi.org/10.1016/j.eswa.2020.113873>
- Turkoglu, B., Uymaz, S. A., & Kaya, E. (2023). Chaos theory in metaheuristics. *Çinde Comprehensive Metaheuristics* (ss. 1-20). Elsevier.
- Wang, G.-G., Guo, L., Gandomi, A. H., Hao, G.-S., & Wang, H. (2014). Chaotic krill herd algorithm. *Information Sciences*, 274, 17-34.
- Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1), 67-82.
- Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., & Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1), 1-37. <https://doi.org/10.1007/s10115-007-0114-2>
- Yang, X.-S. (2020). *Nature-inspired optimization algorithms*. Academic Press.
- ZAIMOGLU, E. A., Celebi, N., & Yurtay, N. (2020). Binary-Coded Tug of War Optimization Algorithm for Attribute Reduction Based on Rough Set. *Journal of Multiple-Valued Logic & Soft Computing*, 35.
- Zawbaa, H. M., Emary, E., & Grosan, C. (2016). Feature Selection via Chaotic Antlion Optimization. *PLOS ONE*, 11(3), e0150652. <https://doi.org/10.1371/journal.pone.0150652>
- Zhang, H., Liu, D., & Wang, Z. (2009). *Controlling chaos: Suppression, synchronization and chaotification*. Springer Science & Business Media.
- Zhao, Y., Dong, J., Li, X., Chen, H., & Li, S. (2022). A binary dandelion algorithm using seeding and chaos population strategies for feature selection. *Applied Soft Computing*, 125, 109166. <https://doi.org/10.1016/j.asoc.2022.109166>

## ÖZGEÇMİŞ

Ad-Soyad : Esin Ayşe ZAIMOĞLU

### ÖĞRENİM DURUMU:

- **Lisans** : 2007, Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği
- **Yüksekisans** : 2018, Sakarya Üniversitesi, Bilgisayar Ve Bilişim Mühendisliği Anabilim Dalı, Bilgisayar Ve Bilişim Mühendisliği

### MESLEKİ DENEYİM VE ÖDÜLLER:

- 2008-2010 yılları arasında AXIS Yazılım Bilişim şirketinde MS Dynamix AX Developer olarak çalıştı.
- 2010-2013 yılları arasında Sakarya Valiliği İl Afet ve Acil Durum Müdürlüğü'nde çalıştı.
- 2013-2015 yılları arasında Selçuk Üniversitesi'nde çalıştı.
- 2015 yılından bu yana Sakarya Üniversitesi'nde Mühendis olarak çalışmaya devam etmektedir.

### TEZDEN TÜRETİLEN ESERLER:

- Zaimoğlu, E. A., Yurtay, N., Demirci, H., Yurtay, Y (2023). A binary chaotic horse herd optimization algorithm for feature selection. Engineering Science and Technology, an International Journal, ISSN 2215-0986. <https://doi.org/10.1016/j.jestch.2023.101453>
- Zaimoğlu, E. A., Celebi, N., & Yurtay, N. (2020). Binary-Coded Tug of War Optimization Algorithm for Attribute Reduction Based on Rough Set. Journal of Multiple-Valued Logic & Soft Computing, 35.
- Zaimoğlu, E. A., Çelebi, N., & Yurtay, N. (2019, July). An Intelligent Feature Selection Method for Finding Rough Set Reducts Based on Tug of War Optimization Algorithm. In International Conference on Intelligent and Fuzzy Systems (pp. 1225-1234). Springer, Cham.
- An Intelligent Feature Selection Method for Finding Rough Set Reducts Based on Tug of War Optimization Algorithm, Zaimoğlu Esin Ayşe, Çelebi Numan, Yurtay Nilüfer, Yayın Yeri: Springer, Editör: Janusz Kacprzyk, Basım sayısı: 1, Sayfa sayısı: 1386, ISBN: 1234-5678,

## **DİĞER ESERLER:**

- ZAIMOĞLU, E., YURTAY, N., & YURTAY, Y. (2018). Analysis Of Student Professions Graduated From Computer Engineering Through Social Networks Using Data Mining Techniques. Presented at the International Educational Technology Conference
- Demirci, H., Yurtay, N., Yurtay, Y., & Zaimoğlu, E. A. (2022). Electrical Search Algorithm: A New Metaheuristic Algorithm for Clustering Problem. *Arabian Journal for Science and Engineering*, 1-20.
- Yurtay, Y., Yurtay, N., Demirci, H., Zaimoglu, E. A., & Göksu, A. (2023). Improvement and Implementation of Sustainable Key Performance Indicators in Supply Chain Management: The Case of a Furniture Firm. *IEEE Access*.